Niko Kivikko

Testidatan hallinnan toteuttaminen GDPR-vaatimusten mukaisesti

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikka
Insinöörityö
15.4.2018

Metropolia

Tiivistelmä

Tekijä(t) Otsikko Sivumäärä Aika	Niko Kivikko Testidatan hallinnan toteuttaminen GDPR–vaatimusten mukaisesti 37 sivua + 3 liitettä 15.4.2018
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori, Simo Silander CTO, Marko Klemetti

Testidata on rajattu määrä tietoja, joiden tarkoitus on toimia sisältönä ohjelmistotestauksessa. Testidataa tarvitaan tekemään testeistä kattavampia ja luotettavampia, jolloin käytettäväksi soveltuva data on yleensä generoitua dataa tai anonymisoitua tuotannon dataa. Testidatan hallinta on testaukseen käytettävien datan tuottamista, jakamista ja varastointia.

Insinöörityössä tarkastellaan GDPR-vaatimukset täyttävää ratkaisua testidatan hallintaan yrityksille maasta ja koosta riippumatta. Euroopan unionin tietosuoja-asetus GDPR tuo tiukennuksia yrityksille, jotka toimivat Euroopan unionin alueella tai joilla on Euroopan unionin jäsenmaissa asuvien henkilötietoja. Yrityksien on asetuksen mukaan saatava datan hallintansa vaatimukset täyttäviksi ennen asetuksen voimaantuloa 25. toukokuuta 2018.

Työssä selvitettiin testidatan hallinnan muutosta historiasta nykyhetkeen. Kuinka paljon testidatan hallinta on muuttunut? Kuinka paljon yritykset ovat saaneet hyötyä kehittämällä testidatan hallintaa ja kuinka paljon kehittäminen tulee tuomaan tietoturvaa jatkossa?

Käytännönosuus sisältää yksinkertaisen esiselvitystyön tekemisen käyttäen Enterprise-tason maksullista ohjelmaa CA Technologies Oy:ltä. Ohjelman nimi on Test Data Manager, joka on heidän tuotteensa testidatan hallinnan saamiseksi GDPR-vaatimuksien mukaiseksi. Esiselvitystyössä käytiin perustaidot datan hallinnasta, mikä kattaa datan generoinnin, datan maskauksen ja datan visualisoinnin. Työssä havainnollistettiin, kuinka datan hallinnointi voidaan tehdä ilman näkyvyyttä tuotantopalvelimelle, joka mahdollistaa henkilötietojen käsittelyn turvallisesti.

Lopputuloksena syntyi suuntaa antava esimerkki testidatan hallinnasta käytännössä ja siitä mitä mahdollisuuksia testidatan hallintaan panostaminen voi yritykselle tuoda. Työn tuloksena nähdään myös, kuinka yritys voi hallita testidataansa täyttäen samalla GDPR-vaatimukset ja välttää Euroopan unionin langettamat sanktiot.

Avainsanat	CA Test Data Manager, MSSQL, Windows Server 2016

Metropolia

Abstract

Author(s) Title Number of Pages Date	Niko Kivikko Test Data Management Solutions for GDPR Compliancy 37 pages + 3 appendices 15 April 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Simo Silander, Senior Lecturer Marko Klemetti, CTO

Test data is a selection of data that have purpose to make software testing coverage larger and testing more reliable. Test data can be made with generating data or masking data from production database. Test data management is handling, sharing and restoring test data.

This thesis focuses on finding GDPR compliant solutions for all sized companies test data management. On 25th of May 2018 GDPR which is European union's data protection regulation comes to effect. All the companies which work in a country that is a member of the European Union or work with the personal data of a European union's citizens, must be GDPR compliant before the regulation comes effect.

In this thesis, we investigate test data management change from history to present and try to find out how much companies got value out of investing to test data management. We are going to peek how secure handling of a personal data will be after making test data managing GDPR compliant and what are the changes that make it possible.

As an example, we will develop a simple test data management solution using Test Data Manager software which is a CA Technologies solution for making test data management GDPR compliant. In this example process we will generate data, mask data and visualize it and that will show us how to manage data without visibility for production database.

The result will introduce an example for test data management in practice and some perspective for how much value investing to the test data management can bring to the company. The result will show how the company can handle their production data with GDPR compliance and avoid the European union's penalties.

Keywords	CA Test Data Manager, MSSQL, Windows Server 2016

Sisällys

Liitteet

Liite 1. Tietokannan SQL-skripti

Liite 2. Käyttäjähallinnan käyttöliittymäkuva

Liite 3. Projektin luonnin käyttöliittymäkuva

Lyhenteet

TDM	Test Data Manager. CA Technologiesin tarjoama testidatan hallinnan työkalu
DBA	Database administrator. DBA on henkilö, joka on yrityksessä vastuussa kaikista tietokannoista ja niiden hallinnasta. Tehtäviin sisällytetään myös testidatan hallinta ja jakaminen kehittäjille.
DevOps	Developers/Operations. Toimintamalli, jossa kehittäjät ja palvelinylläpito tekevät tiivistä yhteistyötä.
GDPR	General data protection regulation. Euroopan unionin tietosuoja-asetus.
Direktiivi	Euroopan unionissa käytössä oleva yleissitova päätös, jota jäsenmaat voivat soveltaa oman maansa lainsäädäntöön sopivaksi.
DPO	Data Protection Officer. GDPR-vaatimuksesta tuleva tietosuojavaltuutettu, joka huolehtii yrityksen tietojen käsittelystä lakien mukaisesti.
OECD	Taloudellisen yhteistyön ja kehityksen järjestö.
Maskaus	Datan muuttamista niin, ettei datasta voi tunnistaa alkuperää.
POC	Proof of concept. Todistus toimivasta ratkaisusta kohdeympäristössä.

1 Johdanto

Tietotekniikan yritysten kilpailu kovenee päivä päivältä, ja kehitys koodin tuottamisen suhteen kiihtyy samassa suhteessa. Ohjelman koodissa ja rakenteessa olevat virheet halutaan löytää mahdollisimman nopeasti, jonka avuksi syntyi aikanaan monia erilaisia testaustapoja. Ohjelman testaukseen lopulta tarvitaan dataa, jonka hankinta toteutettiin aluksi vain kehittäjien tekemillä väliaikaisilla datoilla. Tällaisissa tilanteissa testeistä ei kuitenkaan tullut tarpeeksi kattavia, minkä seurauksena testauksessa alettiin käyttämään tuotannon dataa testien datana.

Tuotantodatan käytön myötä testauksesta tuli villi länsi, eikä kukaan yrityksissä ollut enää varma, missä kaikkialla tuotannon data kulkee. Henkilötietoja saatettiin antaa useiden tiimien testauksen tueksi, eikä kukaan voinut olla enää varma, onko kaikki data tietoturvallisesti hallussa. Asiakkaiden henkilötietojen tietoturvan takaamiseksi alkoi henkilötietojen maskaaminen. Maailmaan kehittyi uusi termi: testidatan hallinta test data management, joka kuitenkaan ei nostanut yritysten kiinnostusta sen ansaitulle arvolle. Halu saada tietoa datan kattavuudesta ja varastoinnista kehitti testidatan hallinnan sellaiseksi, jona me sen tunnemme tänäkin päivänä. [1.]

Ongelmat eivät kuitenkaan olleet yritykselle itselleen kovinkaan haitallisia, ja testidatan hallinnan oikeaoppinen toteuttaminen olisi aiheuttanut yrityksille lisää kuluja, minkä seurauksena useat yritykset eivät siihen juurikaan panostaneet. Pahimmillaan yrityksillä ei ollut edes henkilökuntaa, joka vastaisi yrityksen datasta. Kehittäjät pystyvät tällaisessa tilanteessa hakemaan dataa täysin mielivaltaisesti.

Euroopan unioni aloitti tammikuussa 2012 kehittämään uutta asetusta kansalaistensa tietoturvaa kohentamaan. General data protection regulation GDPR oli ensimmäinen merkittävä vastaisku testidatan hallinnan mielivaltaista hallintaa vastaan. Huhtikuussa 2016 Euroopan unioni äänesti, että GDPOR toteutetaan, ja se tulee voimaan toukokuussa 2018. Tämä oli kova kolaus yrityksille, koska testidatan hallinnan järjestäminen olisi siitä lähtien pakollista rangaistuksen uhalla. Sakot rikkomuksista asetettiin kymmeniin miljooniin euroihin, millä Euroopan unioni mahdollisti, ettei mikään yritys pystyisi maksamaan itseään ulos vastuusta. [4.]

Insinöörityön tavoitteena oli löytää parhaimmat toimintatavat toteuttaa GDPR-asetuksen kattava testidatan hallinta huolimatta siitä, mikä lähtötilanne yrityksellä on. Insinöörityössä käydään läpi muutamia eri tapoja hoitaa testidatan hallintaa. Lisäksi läpi käydään myös eri toimintatapojen tärkeys GDPR-asetuksen kannalta.

Käytännönosuus työstä on tehty esiselvitystyöksi Eficode Oy:lle, minkä tarkoitus on olla ensimmäinen havainnollistava vaihe testidatan hallinnasta GDPR-vaatimuksien mukaisesti.

2 Testidatan hallinta ohjelmistokehityksessä

2.1 Testidatan hallinnan historia

Testauksen historia yltää 1950-luvulle, josta se on kehittynyt aivan 2010-luvulle saakka. Aluksi testauksen ja virheenkorjauksen välille ei tehty minkäänlaista eroa, vaan testauksella haettiin pääsääntöisesti virhetiloja, jolloin kaikki testaus oli suurimmalta osin virheenkorjausta. Kun testauksessa alettiin ottamaan huomioon myös väärien toiminnallisuuksien testausta, tuli testidatan hallinta kysymykseen.

Testidatan käytön ensimmäisiä askelia oli tehdä täysin geneeristä dataa, jonka avulla testattiin, toimiiko ohjelma täysin suunnitellun mukaisesti. Geneerinen data tuotettiin kehittäjien omilla skripteillä, joiden sisältö ja kattavuus vaihteli hyvin paljon kehittäjien välillä. Geneerisen datan ongelmana oli kokonaisuudessaan se, ettei kyseinen data tulisi vastaamaan oikeaa tuotannon dataa.

Testidatan tarve vastata tuotannon dataa päätettiin korjata ottamalla suora kopio tuotannon tietokannasta ja testata sitä vasten uusien ohjelmien toimintaa. Tällöin testidata on aina verrattavissa tuotannon dataan, koska sehän todella on tuotannon dataa. Mikä sitten on ongelmana tuotantodatan käytössä? Tuotannosta löytyy yleisesti henkilötietoja ja arkaluonteisia tietoja kuten sairaushistoriaa, tilaushistoriaa tai rikosrekisteritietoja. Tuotantodatan ongelmana onkin juuri se, että se on tuotannon dataa.

Tuotantodatan oikeaoppinen hallinta syntyi juuri eri historian vaiheiden pohjalta. Osa datasta on syytä generoida. Arkaluonteista tietoa voidaan ottaa suoraan tuotannosta, ja arkaluonteiset tiedot on syytä maskata. Lopulta on syytä vahtia, missä dataa menee ja missä muodossa kyseinen data on. [1.]

2.2 Testidatan hallinnan edut

Testidatan hallinnan ymmärtäminen on usein vaikeaa ja sitä pidetään kehittäjien keskuudessa turhana sijoituksena tai jossain määrin tekijänä, joka hankaloittaa työntekoa. Mielipide ei ole aina täysin väärässä, mutta näkökulma on usein liian suppea. Kattavaa testidataa voidaan generoida suoraan kehittäjien toimesta, jos dataa generoiva kehittäjä tuntee ohjelman kaikki osat, tuntee tuotannon käyttämän tietokannan taulujen skeemat, tuntee tuotannon datan moninaisuuden, osaa säilöä generointiin käytetyt skriptit ja kerätyt testidatat huolellisesti ja osaa määrittää kaikkeen edellä mainittuun kattavat ohjeet tulevia työntekijöitä varten.

Kun kaiken työn tekemisen lisäksi kehittäjä pystyy kehittämään kattavan testidatan, on kehittäjällä oltava paljon organisointikykyjä ja epäinhimillisen kovat hermot. Ohjelmistojen koon kasvaessa myös datan määrä kasvaa, jolloin alkaa olla täysin mahdotonta pysyä kaiken yllä mainitun perässä. Ohjelmiston osia on pakko jakaa useiden tiimien kesken eikä generoitu data enää olekaan synkronoitua toisten tiimien generoidun datan kanssa. Usein kehittäjillä ei saa olla pääsyä tuotannon tietokantoihin ollenkaan, jolloin myös tietokantojen taulujen syntaksit tai datan moninaisuus ei ole täysin hallussa. Tämä johtaa siihen, ettei testidata ole tarpeeksi laadukasta saavuttamaan tarvittavaa kattavuutta.

Tehokkuus tuo rahaa yritykselle, ja siitä syystä isot organisaatiot ovat sijoittaneet suuriakin summia testidatan hallinnan kehittämiseen. Kun saadaan ohjelmistokehittäjät keskittymään ohjelmistojen kehittämiseen ja testaajat keskittymään testaamiseen, testauksen datasta voi huolehtia yrityksen DBA. Laadukasta ja synkronoitua dataa voidaan näin saattaa testaajien käyttöön hyvinkin nopeasti, eikä mikään osa ohjelmistotuotannosta menetä tehokkuuttaan. [2.]

2.2.1 Lait ja säädökset

Huomioon on otettava myös laillinen perspektiivi, jota voidaan havainnollistaa seuraavanlaisilla esimerkkitilanteilla. Ensimmäisessä esimerkissä asiakas maksaa vakuutuksensa laskun verkossa, mikä tapahtuu vakuutusyhtiön tilaamalla maksupalvelulla. Maksun siirrossa tapahtuu virhe, jonka seurauksena asiakkaalta laskutetaan laskun summa kahdesti. Asiakas soittaa virheen huomattuaan vakuutusyhtiön asiakaspalveluun, joka aloittaa selvitysprosessin.

Vakuutusyhtiö tilasi ohjelman alihankintana, jolloin virheiden korjaaminen omalla kehitystiimillä ei tule kysymykseen. Vakuutusyhtiö soittaa maksupalvelun tarjonneelle alihankkijalle ja välittää asiakkaan reklamaation sinne. Vakuutusyhtiön on jollain tavalla saatava selvyys virheen aiheuttajasta, mutta heillä ei ole pääsyä vakuutusyhtiön tietokantaan. Vakuutusyhtiöllä ei ole oikeutta jakaa asiakkaan henkilötietoja eteenpäin, koska kyseessä on arkaluontoisia vakuutusasiakirjoja koskevien laskujen tietoja. Vakuutusyhtiön DBA voi tässä tapauksessa tehdä nykyisten ja haluttujen datojen pohjalta testidatasetin, jonka avulla maksupalvelun tarjoaja pystyy kehittämään tuotteitaan. [2; 5; 4.]

Toisessa esimerkissä ongelmiin törmää suuri kansainvälinen yritys. Palvelua tarjotaan Eurooppaan, asiakaspalvelu sijaitsee Yhdysvalloissa, kehitys Intiassa ja testaus Kiinassa. Euroopan markkinoilla oleva tuote vaatii lisää kehitystä ja ongelmista soitetaan Yhdysvaltoihin. Asiakaspalvelu tekee ratkaisunsa ja lähettää muutokseen tarvittavat tiedot Intiaan, jossa muutokset tehdään. Muutettu tuote matkaa Kiinaan testattavaksi, josta valmis hyväksytty tuote palaa takaisin Eurooppaan. Tiedon siirtoa rajoittaa tässä tapauksessa neljän eri alueen lainsäädäntö, eikä monissa näistä alueista saa missään tapauksessa lähettää henkilötietoja alueen ulkopuolelle. Tällöin ei ole muuta mahdollisuutta kuin turvautua oikeaoppisen testidatan hallinnan työkaluihin. On generoitava sekä maskata testidataa, jonka voi lähettää turvallisesti rajojen ulkopuolelle. [2; 5; 4.]

2.2.2 Testidatan hallinnan edut kehittäjille

Kehittäjien kannalta testidatan hallinta vapauttaa resursseja ohjelmiston kehittämisen ulkopuolelta olevista työtehtävistä. Esimerkiksi kehittäjä tekee muutoksen, joka vaatii testausta. Muutoksessa käytetään samaa dataa, jota käyttää kolme muuta kehitystiimiä. Kehittäjän yrityksessä ei ole toteutettu keskitettyä testidatan hallintaa, josta syystä kehittäjien on generoitava testidata itse. Kehittäjä generoi itse testidatan, jolla hänen testit menevät onnistuneesti läpi. Kaksi muuta kehitystiimiä käyttää omia generaituja datojaan, jotka eivät vastaa kehittäjän dataa. Tuotteen osat yhdistetään, ja data aiheuttaa konflikteja. Tuotteen korjaus aloitetaan ja deadline ylitetään.

Jos kehittäjä tietää datan tarpeesta olla synkroninen myös kahden muun kehitystiimin kanssa, hän ei generoi dataansa itse. Kehittäjä käy tarkistamassa muiden tiimien tilanteen ja toteaa toisen tiimin olleen jo valmis kaksi päivää sitten. Viimeisin tiimi on jäänyt jälkeen sairastapauksen vuoksi ja on täten vasta neljän päivän päästä valmis. Kehitystiimit joutuvat lopulta odottamaan kaikkien tilanteen päätymistä samaan pisteeseen, jolloin voidaan määrittää tarvittavan testidatan generoinnin vaatimukset. Lopulta joku tiimeistä generoi testidatan, ja testaus voi alkaa. Vaihtoehtoisesti jo olemassa olevasta tuotannon tietokannasta voidaan ottaa dataa testaukseen, mutta henkilötietojen kohdalla se ei välttämättä ole sallittu ratkaisu. Kehittäjä voi myös tuotannon dataa maskaamalla tuottaa sallittua dataa, mutta vain jos kehittäjällä on pääsy tuotannon tietokantaan.

3 General data protection regulation GDPR

3.1 GDPR-asetuksen kehittäminen

Euroopan unionin edellinen tietosuoja-asetus tuli voimaan vuonna 1995, jolloin direktiivi 95/46/EC valmistui ja tuli voimaan. Tietosuoja-asetuksen päivityksestä tehtiin ehdotus vuonna 2012 tammikuun 25. päivä Euroopan komission toimesta. Ehdotuksen pohjalta asetusta valmisteltiin kahden vuoden ajan ja vuoden 2014 maaliskuussa Euroopan parlamentti hyväksyi ehdotuksen sellaisenaan. [4.]

Asetuksen tarpeellisuus huomioitiin Euroopan unionissa, ja asetuksen eteneminen oli erittäin nopeaa. Jo kesäkuussa 2015 ehdotus hyväksyttiin sellaisenaan viimeiseen vaiheeseen, trialogineuvotteluihin. [4.]

Trialogi-neuvotteluissa asetuksen sisältöä hiottiin jopa kuuden kuukauden ajan, jonka jälkeen joulukuussa 2015 parlamentti ja valtuusto olivat yksimielisiä asetuksen virallisesta ja lopullisesta sisällöstä. Lopullinen hyväksyntä tehtiin tammikuussa 2016 niin Euroopan parlamentissa kuin Euroopan unionin valtuustossa. Asetuksen täytäntöönpano tulisi olemaan kahden vuoden kuluttua, toukokuussa 2018. [4.]

3.2 Miksi GDPR?

Aiempi asetus 95/46/EC ja GDPR liittyvät vielä huomattavasti vanhempaan ryhmään asetuksia, jonka vuoksi asetuksen päivitys oli täysin tervetullut uudistus Euroopan unionin kansalaisten suojaksi. Taloudellisen yhteistyön ja kehityksen järjestö OECD julkaisi ohjeistuksen henkilötietojen yksityisyyden suojaamisesta, joka viittasi jo vuonna 1980 julkaistuun asiakirjaan henkilötietojen suojaamisesta. Ohjeistus oli kokoelma Euroopan unionin ja Yhdysvaltojen kannattamia suosituksia henkilötietojen suojaamisesta ja keskeisistä ihmisoikeuksista yksityisyyteen liittyen. [3; 4; 5.]

OECD:n ohjeistuksen mukaan yritys saa kerätä vain lain ja tarkoituksen mukaista dataa sekä vain tarkoituksen mukaisen määrän. Datan tulee myös olla tarkoitukselle relevanttia ja ajan tasalla olevaa. Tarkoitus datan käytölle tulee olla hyvin perusteltua, eikä sitä tule käyttää mihinkään muuhun tarkoitukseen ilman asianomistajan tai viranomaisten lupaa. Data tulee suojata turvallisuusriskejä vastaan kuten hävittämistä, luvatonta pääsyä, tuhoutumista, luvatonta käyttöä, luvatonta muokkaamista ja julkituloa vastaan. Henkilötietojen omistajalla pitää olla helppo pääsy käsiksi omiin tietoihinsa, tieto kenellä kyseistä tietoa on ja ketä sitä käyttää. Kaikkien henkilötietoja käsittelevien tahojen on sitouduttava noudattamaan näitä sääntöjä, vaikka tämän tyyliset säädökset oli monen maiden lakiin lisätty jo ennen ohjeistuksen julkaisua. [3; 4; 5.]

EU Direktiivi 95/46/EC

Euroopan unioni ei direktiivissään tuonut ohjeistukseen juurikaan mitään uutta, eikä muuttanut mitään vanhaa. Kaikki OECD:n ohjeistuksessa tulleet asiat pysyvät koskemattomina. Direktiivin suurin uutuus ja syvin tarkoitus olikin vahvistaa ohjeistuksen noudattamista Euroopan unionin alueella sekä varmistaa mahdollisimman hyvä turva myös Euroopan kansan datalle Euroopan unionin alueiden ulkopuolella. Jokaiseen Euroopan unionin jäsenmaahan nimitettiin viranomaisia (DPA, Data Protection Authorities) valvomaan direktiivin noudattamista ja hyväksyi henkilötietojen luovutusta kolmansiin maihin koskevia lupia yrityksille. [4; 6.]

3.3 Mitä GDPR pitää sisällään?

Datanhallinnan rikkoutumisesta ilmoittaminen

GDPR tuo mukanaan ilmoitusvastuun yrityksille datan hallinnan rikkomuksista. Kun tietovuoto tai muu dataan liittyvä rike havaitaan, yrityksellä on kolme vuorokautta aikaa ilmoittaa siitä asiakkailleen ja valvontaviranomaisille datarikkomuksen selvittyä. [3; 4; 5.]

Oikeus tulla unohdetuksi

Asiakkaalla on oikeus vaatia henkilötietojensa poistoa kaikkialta yrityksen järjestelmistä. Tämä koskee myös kaikkea alihankkijoille ja työntekijöille päätynyttä henkilötietodataa. Asiakkaan vaatimusta henkilötietojensa poistamisesta pitää kuitenkin peilata maan omaan lainsäädäntöön, minkä seurauksena lain määritteleminen jää lain tulkinnan varaan. Asetus ei ole vielä tullut voimaan. Tästä syystä siitä ei ole ennakkotapausta. [3; 4; 5.]

Oikeus päästä käsiksi omiin tietoihin

Asiakkaalla on oikeus vaatia omia tietojaan yrityksen järjestelmästä, eikä asiakkaalla ole tarvetta kertoa, minne tai mihin tarkoitukseen tietonsa tarvitsee. Tällä asetuksella tuodaan lisää läpinäkyvyyttä yrityksen dataan ja sen datan hallintaan. [3; 4; 5.]

Datan siirto

Asiakkaalla on oikeus vaatia omien tietojen siirtoa yrityksen tietokannasta toiseen yritykseen. Datan täytyy olla yleisesti käytetyissä ja koneella luettavissa olevissa muodoissa. [3; 4; 5.]

Sisäänrakennettu yksityisyydensuoja

Sisäänrakennettu yksityisyydensuoja ei ole mikään uusi keksintö. Se on ollut osana lain sääntöjä monissa maissa ja myös direktiivissä 95/46/EC. On ymmärrettävää kyseisen hyväksi todetun osan liittäminen myös GDPR:n piiriin, mutta uutena saavutuksena tämä kyseisen osan noudattaminen tulee olemaan myös selvästi rangaistuksen uhalla noudatettava. Sisäänrakennetulla yksityisyydensuojalla tarkoitetaan datan hallinnan rakennetta, missä datan käytöstä vastaa jokin henkilö, joka jakaa dataa vain sitä tarvitseville, sekä valvoo että dataa käytetään juuri siihen tarkoitukseen johon sitä on tarvittu. [3; 4; 5.]

Tietosuojavaltuutettu (DPO)

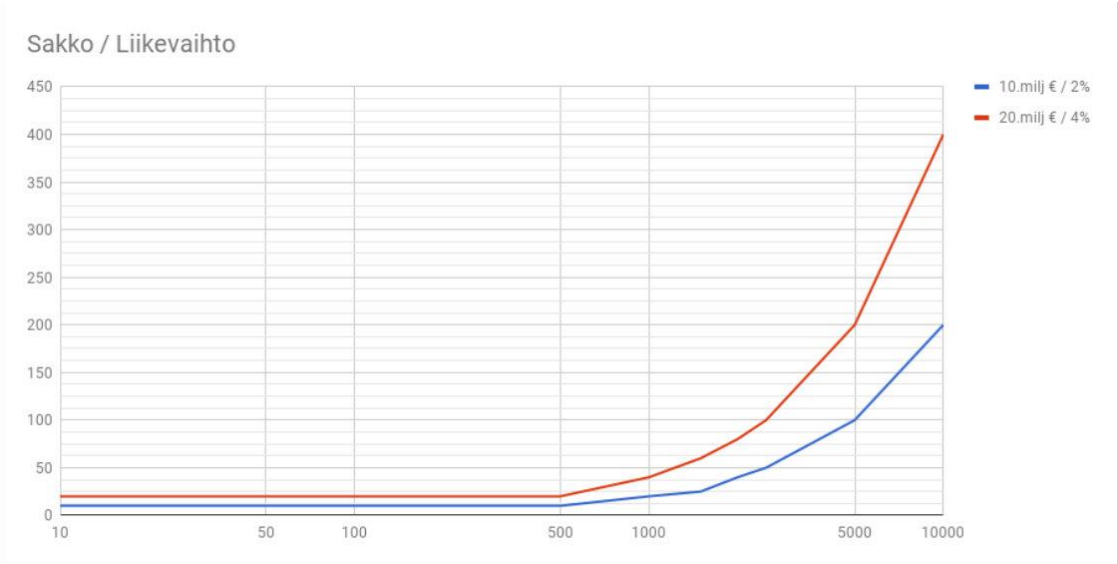
Asetuksen mukana tulee myös tarve asettaa vastuuhenkilö yrityksen sisälle valvomaan GDPR:n mukaan toimimista. Tämä henkilö tulee vastaamaan Euroopan unionille kaikesta GDPR:n liittyvästä sekä antaa ilmoitukset mahdollisista datarikkomuksista. Tietosuojavaltuutetun tulee olla työtehtäväänsä pätevä, omata erinomaista tietoa tietosuoja lainsäädännöstä ja toimintatavoista. Tietosuojavaltuutettu voi olla joko sisäistä henkilöstöä tai alihankkija. Tietosuojavaltuutetun yhteystiedot on annettava alueella vaikuttavalle valvontaviranomaiselle. Tietosuojavaltuutetulle pitää tarjota työntekoon ja tietotaitonsa ylläpitoon tarvittavat resurssit. Tietosuojavaltuutetulla pitää aina olla suora yhteys yrityksen ylimpään johtoon, minkä avulla ilmoittaa mahdollisista rikkomuksista. Tietosuojavaltuutetulla ei saa olla muita työtehtäviä, jotka voivat saattaa tietosuojavaltuutetun ristiriitaan säädösten noudattamisen ja tuloksen saavuttamisen välillä. [3; 4; 5.]

Rangaistukset

Euroopan unionin GDPR-asetuksessa suurimpana muutoksena verrattuna 95/46/EC-direktiiviin ovat rangaistukset. Kun 95/46/EC-direktiivi ei määrittänyt tarkkaan kuinka asetusta kohtaan tehtyjä rikkomuksia tulisi käsitellä, on GDPR:llä erittäin tarkat ohjeet, kuinka eritasoisista rikkomuksista tulee rangaista.

Lievempänä rangaistuksena yritys voi saada kirjallisen varoituksen ja Euroopan unioni sijoittaa yrityksen kausiluontoiseen tehovalvontaan. Varoitus on lievissä tilanteissa varmistus sille, etteivät rikkomukset jatku ja korjaustoimenpiteisiin aletaan välittömästi. Tehovalvonnan kautta saadaan nopeasti kiinni yritykset, jotka eivät tee tarvittavia parannuksia ja rangaistusta voidaan koventaa.

Sakoista lievempi on 10 miljoonaa euroa tai kaksi prosenttia yrityksen liikevaihdosta riippuen siitä, kumpi on suurempi. Sakkorangaistukset on suunniteltu annettavaksi aina, jos huomataan yrityksen olevan haluton rakentamaan datan hallintaansa kattamaan asetuksen vaatimia asioita. Pienemmällä sakolla selviää, jos todetaan, että datan hallinnassa löytyy puutteita, jotka ovat olleet yrityksen tiedossa, mutta niiden korjaamista ei ole edes suunniteltu. Sakkoon päädytään myös tilanteissa, joissa yritys ei huomioi valvontaviranomaisen kirjallista varoitusta tai yritys yrittää tahallaan vaikeuttaa valvontaviranomaisten työtä. [3;4;5.]



Kuva 1. Kaavio sakkorangaistuksen suuruudesta verrattuna liikevaihtoon. [4.]

Korkein mahdollinen Euroopan unionin langettama rangaistus on 20 miljoonan euron tai neljän prosentin osuuden liikevaihdosta suuruinen sakko asetusta noudattamattomalle osapuolelle. Sakon suuruus määräytyy aina suurimman mahdollisen euromäärän mukaisesti, jolloin sakko on aina enemmän kuin 20 miljoonaa euroa. Suurimman mahdollisen sakon voi saada osoittamalla täydellistä piittaamattomuutta asetuksen datan hallintaa koskevia sääntöjä kohtaan, kun piittaamattomuuden on todettu aiheuttaneen tai saattavan aiheuttaa suurta vahinkoa Euroopan unionin jäsenmaiden asukkaille, eikä yritys osoita halukkuutta korjata käytäntöjään asetusten mukaisiksi. [4.]

4 Datan hallinnan kokonaiskuva

Yrityksen datan hallinnan rakenne

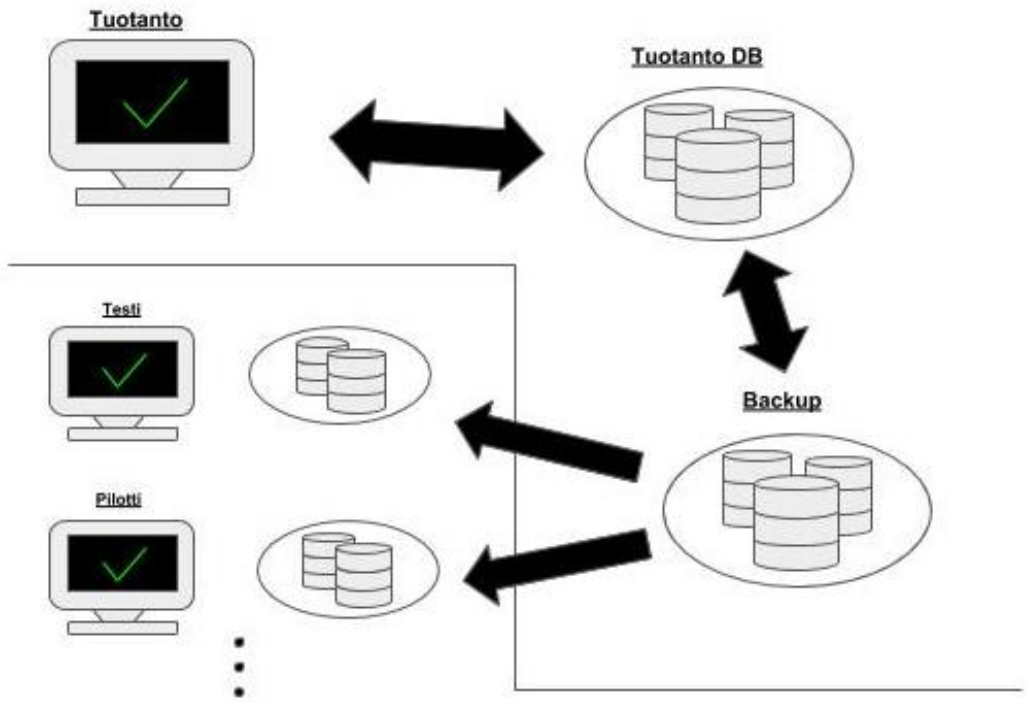
Yritys, jonka tietokantojen kokonaisuus on hyvällä tasolla, pystyy toteuttamaan geneeristä ja maskattua dataa helposti ja nopeasti. Yrityksellä on usein nimitetty DBA , jonka vastuulla on yrityksen datan hallinta sekä testidatan tarjoaminen kehittäjille. DBA hoitaa työnsä joko itse tehtyjen skriptien tai yrityksen hankkiman maksullisen työkalun avulla. Kehittäjillä ei ole pääsyä tuotantoon, eikä tarvetta huolehtia testauksessa käytettävästä datasta. [12.]

Yrityksillä on tyypillisesti oma tuotannon tietokanta, jossa ohjelmiston keräämä ja tarvitsema data sijaitsee. Yrityksen tietokantojen kokonaisuus on kuitenkin huono, jos kokonaisuus on päätetty jättää tälle tasolle. Tuotannon tietokannasta tulisi olla varmuuskopio, joka sijaitsee eri palvelimella kuin tuotannon tietokanta. [12.]

Yrityksen kehitys on tapahduttava omilla palvelimilla, joissa pyörii testidataa. Palvelimia on yleensä enemmän kuin kaksi ja mahdollisesti eri kehittäjillä on samaan ohjelmaan tulevat eri osat työn alla ja sen mukana myös tarve samalle testidatalle. Joissakin yrityksissä on päädytty antamaan kehittäjille vapaat kädet ottaa tuotannon varmuuskopiosta tietokanta-dumppi omalle koneelle, jonka avulla tehdään halutut testit. Kehittäjien päästäminen käsiksi tuotannon tietoihin ei kuitenkaan ole ollut vuosiin sallittua ja Euroopan unionin regulaatioita ei tällöin noudatettu.

Kiinnijääminen ja sanktiot olivat tunnetusti harvinaisia, jolloin henkilötiedot liikkuivat vapaasti ja vastuuttomasti. [12.]

Datan jakamisesta vastaava DBA on oikea henkilö jakamaan testidatat tiimeille, varmistamaan niiden tietojen kattavuus testauksessa ja huolehtimaan, ettei testauksessa käytetä henkilötietoja suoraan sellaisenaan. DBA ottaa varmuuskopiosta tietokanta-dumpin, josta hän maskaa henkilötiedot tunnistamattomiksi, generoi puuttuvaa tietoa ja varmistaa, että kaiken testidatan sijainti tiedetään. [12.]



Kuva 2. Hahmotelma yksinkertaisesta datan hallinnan rakenteesta [12.]

Huomaamme kuvasta 2, minkälainen on yksinkertaisemmillaan datan hallinnan rakenne. Tämän kuvan mukainen rakenne on vähintä, mitä tämän hetken yrityksillä pitäisi jo olla olemassa.

5 Ratkaisu testidatan hallintaan käyttäen Enterprise-ohjelmistoa

5.1 Mikä on CA Technologies ja CA Test Data Manager?

CA Technologies on alun perin amerikkalainen ja myöhemmin kansainvälistynyt yritys, joka perustettiin vuonna 1976. CA Technologies tarjoaa monipuolisesti työkaluja projektin suunnittelusta testaukseen ja siitä jatkuvaan julkaisuun saakka. Yritys on tunnettu tavastaan ostaa kilpailijoitaan pois markkinoilta ja liittää heidän tuotteet omaan valikoimaansa. [11.]

CA Technologies tarjoaa testidatan hallinnalle työkalua nimeltä CA Test Data Manager, jolla tullaan havainnollistamaan testidatan hallinnan kannalta tärkeitä toimintatapoja.
Test Data Manager eli lyhennettynä TDM koostuu useista pienemmistä ohjelmista, jotka on yritysostojen yhteydessä kasattu yhteen. Näitä ohjelmia ovat GT Datamaker-nimisen ohjelman, joka kattaa GTEDI-, GTDiagrammer- ja Test Data Visualizer -nimiset ohjelmat datan hallinnointiin ja havainnollistamiseen. GT Datamakerin lisäksi tuotteeseen on lisätty Fast data masker datan maskaamiseen sekä Javelin, jolla voidaan hallita koko tuotetta. TDM tarjoaa myös web-käyttöliittymän GT Datamakerin palveluille, jolloin pääsyä master-päätelaitteelle ei tarvita esimerkiksi datan haussa. [6.]

5.2 Vaatimukset

TDM vaatii master-päättelaitteelle Windows 7:n tai uudemman käyttöjärjestelmän, RAM-muistia vähintään 3 GB, 20 GB vapaata tilaa kiintolevyltä, vähintään 2 GHz:n prosessorin ja internetyhteyden vähintään 100 Mbit nopeudella. Palvelinvaatimukset jakautuivat kahteen luokkaan: normaali käyttöön, joka kattaa noin viisi samanaikaista käyttäjää, ja vaativaan käyttöön, jossa käyttäjämäärä voi olla huomattavasti tätä suurempi. Insinöörityön esimerkkitapauksessa keskityttiin tarjoamaan palvelua yritykselle, jolle riittää normaali käyttö, jonka vaatimukset ovat Standard Edition Windows Server 2008 tai uudempi käyttöjärjestelmä, 20 GB tyhjää kiintolevytilaa, vähintään yksi neliydinprosessori, 16 GB RAM-muistia ja 1 Gbit:n internetyhteys. Kohteena olevat tietokannat voivat olla, minkä tyyppisiä tahansa. Osaa tietokantatyypeistä tuetaan tuotteen kautta suoraan ja loppujen toiminta voidaan tehdä käyttäen Javelinin tarjoamaa REST-rajapintaa. [6.]

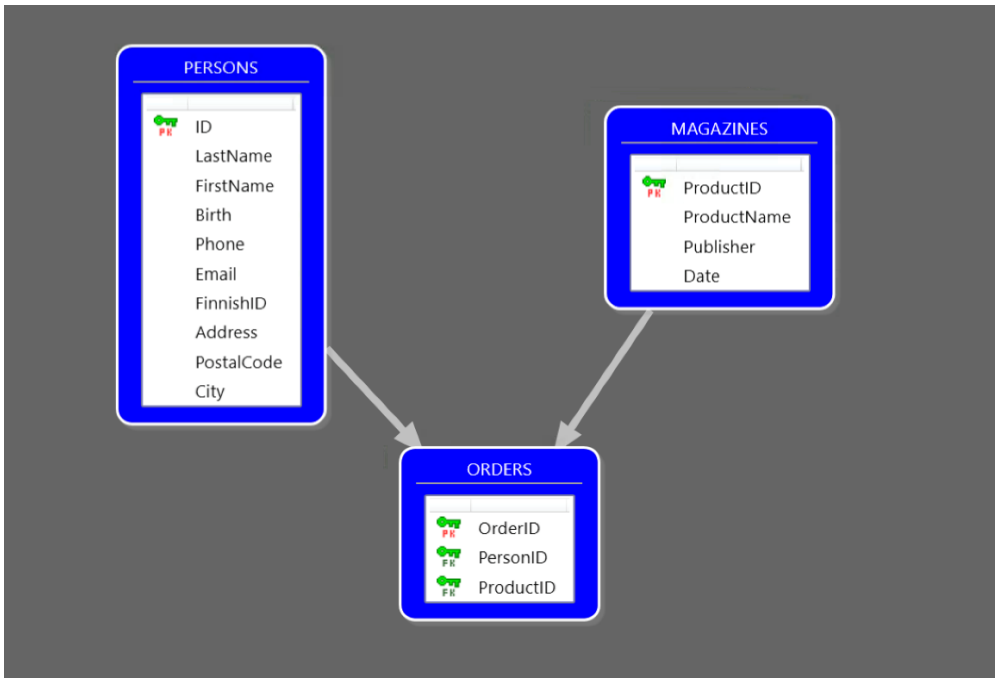
5.3 Tietokantayhteydet

Käyttäjä ottaa yhteyden päätelaitteella TDM-palvelimelle, jonka takana toimii tuotteen toiminnallisuuksiin tarvittava tietokanta. Tietokantaa hyödyntäen tuote pystyy tekemään tarvittavat toiminnot datalle ja saa suoralla yhteydellä lähetettyä tiedot testiympäristöjen tietokantoihin. Jos ei ole mahdollista rakentaa tuotteesta yhteyttä testiympäristöjen tietokantoihin, niin on uusi data vietävä manuaalisesti käyttäen dumppia. TDM-palvelimelta on myös mahdollisuus ottaa yhteydet LDAP- ja Mail-palvelimiin. [6.]

5.4 CA Test Data Manager -ohjelmalla tehty havainnollistava esimerkkitapaus

Esimerkkitapauksen valmistelut

CA TDM asennetaan Windows Server R2 2016 -käyttöjärjestelmälle, joka on vähimmäisvaatimukset täyttävällä palvelimella. Tietokantoja rakennettiin kolme kappaletta, joista kaikki ovat esimerkkitapauksessa MSSQL-tyyppiä. Palvelimelle tehdään kolme instanssia, joista ensimmäinen TDMSQLEXPRESS on TDM-tietokantana toimiva instanssi. Toinen, joka on nimeltään TDMTESTSOURCE, kuvaa tuotannon tietokantaa, josta dataa haetaan. TDMTESTTARGET kuvasi testiympäristön tietokantaa, jonne data täytyi saada siirrettyä.



Kuva 3. Esimerkkitapauksessa käytettävän tietokantarakenteen skeema [8.]

TDMTESTSOURCE-instanssiin tehdään tietokanta test_db1, jossa on taulut MAGAZINES, ORDERS ja PERSONS, kuten voimme kuvasta 3 huomata. Esimerkkitapausta varten tehty kantarakenne kuvasi lehtimyyntiin keskittyneen yrityksen tietokantoja ja niiden riippuvuuksia. Henkilötiedot ovat sijoitettuna tauluun PERSONS, joka tulisi olemaan esimerkkitapauksen kannalta merkityksellisimmän muutoksen kokeva osuus. Data tuotetaan kantaan generoimalla, jolloin voidaan havainnollistaa kaikki GT Datamakerin tarjoamat ominaisuudet.

TDMTESTTARGET-instanssi sisältää kannan test_db2, jossa on sama skeema kuin tuotannossa, mutta ei ollenkaan dataa. Kyseiseen test_db2-kantaan tullaan tuomaan dataa maskaamalla suoran tietokantayhteyden avulla.

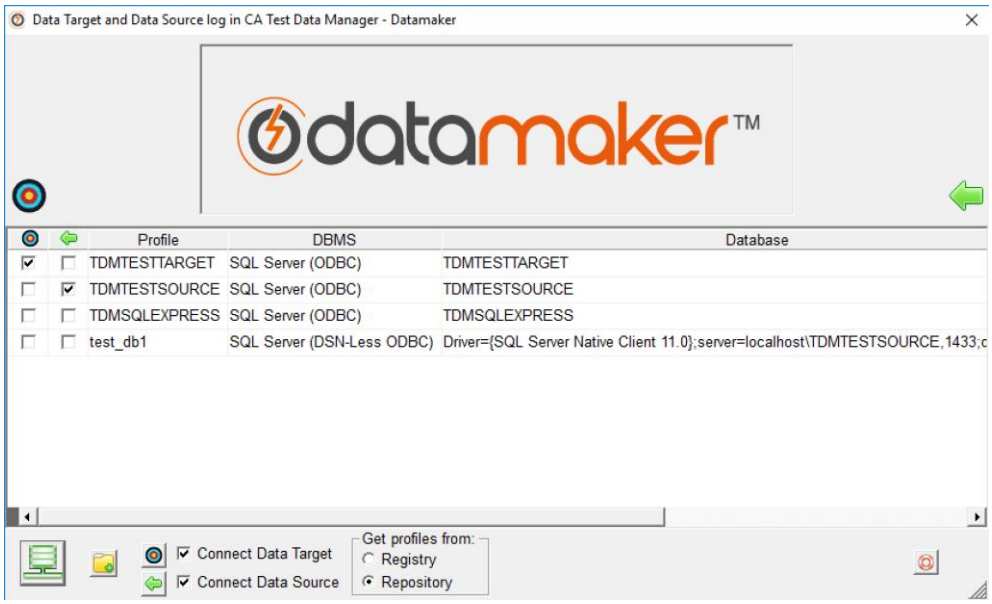
Esimerkkitapauksen ulkopuolelle jätetyt asiat

Havainnollistavassa esimerkkitapauksessa ei käytettä Javelinia tai REST-rajapintaa kokonaisuuksien hallitsemiseen. Työn keskiössä on tällöin yrityksille ajankohtaisimmat ominaisuudet, kuten datan generointi ja maskaus sekä datan turvallinen siirtäminen suorilla tietokantojen yhteyksillä. Esimerkkitapauksessa käytettävän kannan rakenne on yksinkertainen, jolloin useiden testiympäristöjen tietokantoihin ei tultu lisäämään yhteneväistä dataa. [7.]

5.5 Datan generointi

Datan generoinnin valmistelut

Tuotannon tietokantana pystytetään TDMTESTSOURCE-instanssiin kanta nimeltä test_db1. Kyseinen kanta ei sisällä alustavasti muuta kuin skeeman. Datan saamiseksi kyseiseen kantaan on mahdollista käyttää TDM:n tarjoamaa GT Datamaker -työkalua.



Kuva 4. Yhteyksien asettaminen GT DataMaker -ohjelmalle.

Aina Datamaker -ohjelmaa avattaessa on valittava kolme tietokantaa, jotka ovat TDM repository, lähdekanta ja kohdekanta. Lähdekanta on kuvattu vihreällä nuoli-symbolilla ja kohdekanta on kuvattu maalitaulu-symbolilla. Ohjelma listaa kaikki tietokannat, jotka ovat yhdistettyinä TDM -palvelimelle. Tämä mahdollistaa työskentelemisen useiden tietokantojen kanssa organisoidusti. [7.]

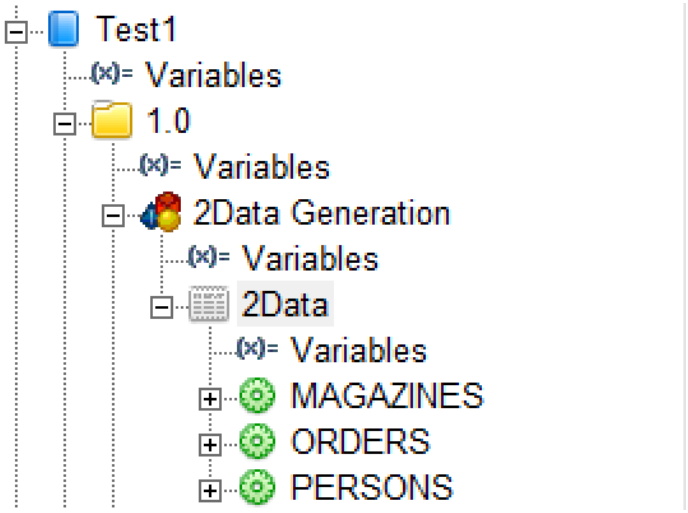
Käyttöoikeuksien hallinta

Ohjelmaa käytettäessä yritysmaailmassa, on hyvä jakaa oikeudet eri tietokantoihin ja admin-oikeuksiin. Esimerkkitapauksessa ei tehdä uusia käyttäjiä, mutta käydään läpi peruskäyttäjähallintaa, joka kattaa uusien käyttäjien lisäämisen sekä oikeuksien antamisen käyttäjille. Security-osasto löytyy vakionäkymästä ja ohjelma kysyy heti painalluksen jälkeen käyttäjähallintaan oikeudet omaavaa käyttäjätunnusta. Yleensä admin-tasoiset käyttäjät tekevät kaiken käyttäjähallinnan. Täältä voidaan tehdä käyttäjiä erilaisilla oikeuksilla ja liittää ryhmiin, joilla on oikeuksia tiettyihin projekteihin (liite 2). Yleensä ryhmiä tehdessä on hyvä jakaa oikeudet projektikohtaisesti eli yhdessä ryhmässä on oikeudet yhteen projektiin. TDM tukee myös jollain tasolla AD- ja LDAP-käyttäjähallintaa, joka on usein yrityksen näkökulmasta haluttu ominaisuus. [7.]

5.5.1 Projektin luonti

Jokainen datan hallinta on projekti, joka luodaan GT Datamakerilla tai verkkopohjaisella TDM portal-palvelulla. TDM ei välitä, millä tavalla projekteja jakaa tai nimeää, mutta on hyvä tapa organisoida projektit vastaamaan kehitysprojektia, jolloin tiedetään, mitä projektia varten testidataa halutaan. Jos kehittäjät ovat tekemässä esimerkiksi uutta verkkomaksupalvelua, jolloin projektin nimi voisi olla verkkomaksupalvelu-cdc. Tällöin myös TDM-projektin nimeksi voidaan antaa verkkomaksupalvelu-cdc, jolloin selkeä yhteys löytyy suoraan nimestä.

Projektin luonti tapahtuu hyvin yksinkertaisesti. Projects-kansiorakenteesta näkyville avautuu valikko, josta uuden projektin voi tehdä (Liite 3.). Projektille riittää yleisimmissä tapauksissa vakiona annettavat perustiedot eli projektin nimi, projektin selite, version nimi ja version selite. Projektille voidaan antaa myös tarkempia tietoja kuten projektin juurikansion sijainti, mutta tässä esimerkkitapauksessa ei mennä niin vaativalle tasolle. Useimmissa testidatan hallinnan tapauksissa kyseisiä tietoja ei tarvitse asettaa. Esimerkkitapauksessa annetaan projektille nimeksi Test1 ja versioksi 1.0. Molemmille tiedoille tulee selitteeksi kopio nimestä, eli versiolle 1.0 ja projektille Test1. Lopuksi GT Datamaker vaatii varmistuksen asetetuista arvoista, koska näitä tietoja ei tässä kyseisessä projektissa pysty enää jälkikäteen muuttamaan. [7.]



Kuva 5. Projektin rakenne.

Kun projekti on luotu, on projektille tehtävä Data Group, Data Set ja Data Pool. Data Group ja Data Set osioiden tarkoitus on olla rakenteita, johon tehdyt Data Pool -kokonaisuudet sijoitetaan. Data Pool pitää sisällään kaikki yksittäisessä datan generoinnissa käytettävät säännöt. Nimeäminen on täysin vapaavalintaista, joten demossa nimeämme Data Groupin 2Data Generation -nimiseksi, Data Set nimetään 2Data ja ensimmäinen Data Pool nimetään tietokantaan tehdyn PERSONS-taulun nimen mukaisesti.

5.5.2 PERSONS-taulu

PERSONS-taulu on ensimmäinen generoitava kohde ja siksi PERSONS data-poolille pitää asettaa sama skeema kun PERSONS-taululla. Projects alta valitaan register ja tämän jälkeen valitaan database table. Ohjelma listaa kaikki käytettävissä olevat taulut listaan, josta etsimme taulun PERSONS, jonka rekisteröimme. Tässä vaiheessa on yksi rekisteröity taulurakenne projektissa.

Datan generointi itsessään tapahtuu käyttäen aikaisemmin tehtyä data poolia, joka nimetään PERSONS-taulun mukaan. Data Pool tarkoitus on pitää sisällään rekisteröityjä kantoja kohtaan tehtävät muutokset. Data poolia muokkaamalla pääsemme valitsemaan muutamasta vaihtoehdosta tavan, jolla dataa halutaan generoida. Tässä esimerkkitapauksessa generoidaan dataa käyttäen manuaalista toimintoa. Valitsemalla PERSONS-taulun vasemmalta laidalta tulee näkyviin rekisteröinnin seurauksena tullut PERSONS-taulun skeema. Kun halutaan generoida dataa ensimmäisen kerran, on lisättävä uusi rivi sääntöpohjaksi, koska vakiona poolissa ei ole sääntöjä entuudestaan. Pakollisia puuttuvia tietoja ohjelma kuvaa punaiseksi väritetyillä lohkoilla, ja keltaiset lohkot kuvaavat puuttuvia tietoja, jotka eivät ole pakollisia. [8.]

ID

Sääntöjen rakentaminen tapahtuu lohkoittain. Ensimmäiseksi rakennetaan PERSONS-taulun ID -rakenteen. ID on taulun PERSONS perusavain, jolloin jokaisessa rivissä ID:n on oltava uniikki. Perustana säännöille tulevat olemaan funktiot, tietokannan sarakkeet ja muuttujat. Tässä esimerkkitapauksessa käydään läpi vain kannalle tarpeelliset

toiminnot, mutta valittavana on myös suuri määrä erilaisia toimintoja sääntöjen rakentamiseen.
ID vaatii jatkuvan numeroinnin, joka toteutetaan käyttäen Next value -muuttujaa. Next value -muuttuja antaa jokaiselle riville uniikin ID:n järjestyksessä alkaen numerosta yksi.

FirstName ja LastName

Seuraavana vuorossa on sukunimi- ja etunimisarakkeiden säännöt. GT Datamaker tarjoaa kattavan listan erilaisia etunimiä ja sukunimiä monista eri maista kuten Ruotsista ja Pohjois-Amerikasta, mutta tässä esimerkkitapauksessa toteutamme tarjontaa kuvitteelliselle suomalaiselle yritykselle, josta syystä tarvitaan listat suomalaisista etunimistä ja sukunimistä. Tämäntyyppisissä tilanteissa on rakennettava oma seed-lista, joka sisältää tarvittavan datan. Sääntö itsessään käyttää kolmea funktiota sisäkkäin. Tarkoituksena on valita sattumanvaraisesti listasta arvo väliltä yhdestä kymmeneen ja työssä käytetään yksinkertaisuuden vuoksi molemmissa samaa ratkaisua. Tietysti oikeassa testidatan hallinnan tilanteessa skaala olisi huomattavasti suurempi ja listassa vaadittaisiin olevan enemmän arvoja. [8.]

Seed List

Listojen tekeminen suoritetaan Tools-valikon kautta ja valitaan uusi seed-lista. Listoille valitaan nimiksi tässä esimerkkitapauksessa Finnish Firstname ja Finnish Lastname. Etunimet ja sukunimet voidaan myös asettaa samaan listaan lisäämällä useampia sarakkeita, mutta työssä todettiin selkeämmäksi pitää ne erillään. Kun lisäykset on tehty yhden sarakkeen osalta, lisätään tämän arvoksi haluamamme arvot, jotka tässä kyseisessä tapauksessa ovat etunimet ja sukunimet. Lista on valmis, kun kaikki halutut arvot on asetettu sarakkeelle.

Birth

PERSONS-taulun ikä sarakkeen tiedoissa halutaan olevan vain täysi-ikäisiä henkilöitä. Tällä saamme kuvaavaa tietoa oikeanlaisesta asiakastietokannoista, ja täten esimerkkitapaus pysyy uskottavana. Sääntö rakennetaan funktiosta @randdate, joka valitsee satunnaisen päivämäärän väliltä 01.01.1960 - 31.12.2000. [8.]

Phone

Esimerkkitapausta varten tarvitaan suomalaisella suuntanumerolla ja operaattoritunnuksella olevia puhelinnumeroita. Tähän tarvitaan uusi seed list, joka sisältää arvot +35840, +35845 ja +35850. Jäljelle jäävät seitsemän numeroa arvotaan satunnaisina käyttäen funktiota @randdigit. Kyseinen tapa generoida puhelinnumeroita saattaa tuottaa kahdelle henkilölle saman numeron, mutta esimerkkitapauksen kannalta tällä ei ole merkitystä. Todellisessa testidatan hallinnan tilanteessa jouduttaisiin tuottamaan lisää ehtoja puhelinnumeroille.

Email

Sähköpostiosoitteiden suhteen on pidettävä mielessä, että yrityksessä on yleisesti käytössä koko nimestä ja yritysdomainista koostuva osoite. Esimerkkitapauksessa rakennetaan osoitteet vastaamaan etunimi.sukunimi@tdm.fi -rakennetta, jossa kuuluisi olla myös vaihtuva sähköpostipalveluntarjoaja, mutta se ei ole esimerkkitapauksen kannalta oleellinen tieto. Osoitteiden sääntö on poikkeava edellisistä säännöistä, sillä siinä ei käytetä ollenkaan funktioita vaan PERSONS-kentän sarakkeita FirstName ja LastName. Säännöksi rakentuu FirstName.LastName@tdm.fi, jolloin generaatio tapahtuu etunimen ja sukunimen mukaan. Tässäkin tapauksessa on ongelma oikeassa yritysmaailman datan generoinnin tilanteessa. Jos yrityksessä on kaksi henkilöä, joilla on sama etu- sekä sukunimi, heille tulee myös sama sähköpostiosoite. Esimerkkitapauksessa ei kuitenkaan mennä niin syvälle, että rakennettaisiin ehdolla eri osoitteita tämäntyyppistä tapausta silmällä pitäen. Tämä on kuitenkin hyvä pitää mielessä, kun ollaan oikeassa datan generoinnin tilanteessa. [8.]

FinnishID

Suomalaisia henkilötunnuksia varten CA technologies on kehittänyt valmiita funktioita rakentamaan uskottavia tietoja. Esimerkkitapauksessa valitaan itse syntymäpäivän mukaisesti generoituja henkilötunnuksia, joista jätin sukupuolen vaikuttavan tekijän pois, koska kyseinen tieto ei ole esimerkkitapauksen kannalta merkityksellinen. Sukupuolen vaikuttava tekijä on saatavilla ehtoon mukaan ja on oikeassa generointi tilanteessa syytä ottaa huomioon.

Address

Osoitetiedot rakennetaan käyttäen valmista seed-listaa, jossa on esimerkkitapausta varten tarpeeksi kattava määrä suomalaisia paikannimiä. Numeroinnin toteuttaminen osoitteille ei ole esimerkkitapauksen kannalta merkityksellistä, mutta numerointi voidaan esimerkiksi suorittaa numerofunktioilla. [8.]

Postalcode

Postinumeroiden määrittäminen tehdään testauksessa tarvittavien vaatimusten mukaisesti. Jos numeroilla ei ole merkitystä, niin viisinumeroisia lukuja tuotetaan sattumanvaraisesti käyttäen valmista funktiota. Jos numeroilla on merkitys, niin on rakennettava seed-lista, jossa sarakkeissa on Suomen kaupunkeja ja arvoina on kyseisen kaupungin postinumeroita. Myös jos haluaa viedä tietojen oikeellisuutta pidemmälle niin myös kadunnimet on sijoitettava niin, että ne löytyvät kyseisestä postinumerosta. Esimerkkitapauksessa tarvitaan vain viisinumeroisia lukuja, joten sääntöksi asetettiin seuraavaksi: numero nolla ensimmäiseksi numeroksi, @randdigit-funktio avulla generoidaan numerot ja ehtona käytetään neljän numeron rajoitusta.

Julkaisu ja todennus

Kun säännöt ovat valmiina, halutaan tehdä julkaisu data targetille. Tämä tapahtuu valitsemalla valikosta Publish to Data Target. Seuraavassa vaiheessa päätettiin, kuinka monta riviä halutaan generoida ja valinta tapahtuu lisäämällä toistoja. Esimerkkitapauksessa halutaan tehdä 500 riviä saadakseen sopivan määrän tietoa seuraavia vaiheita varten. Valinta hyväksytään ja generointi annetulle kohdepalvelimelle alkaa.

Firstname	Lastname	Id	Birth	Finnishid	City	Productname
Sonja	Koivisto	248	1962-03-12	120362-6770	Oulu	Kauppalehti
Liisa	Kivi	492	1998-01-22	220198-035D	Helsinki	Kauppalehti
Jorma	Kivi	331	1989-06-18	180689-722P	Sipoo	TM
Kalle	Noronen	454	1997-06-09	090697-5392	Tuusula	TM
Kaisa	Tammi	404	1961-01-17	170161-321V	Helsinki	TM
Viivi	Virtanen	7	1980-12-18	181280-590Y	Espoo	TM
Villa	Timonen	394	1966-01-08	080166-800T	Turku	TM
Kaisa	Timonen	52	1967-01-19	190167-0565	Oulu	TM
Teemu	Koivisto	215	1993-01-07	070193-1549	Rovaniemi	TM
Teemu	Koivisto	500	1997-09-17	170997-199N	Oulu	TM
Villa	Koivisto	236	1964-12-12	121264-217X	Helsinki	TM
Teemu	Salo	355	1991-01-21	210191-117L	Vantaa	TM
Viivi	Risikko	65	1987-09-27	270987-633J	Espoo	TM
Teemu	Koivisto	500	1997-09-17	170997-199N	Oulu	TM
Vesa	Noronen	375	1966-09-17	170966-0540	Kerava	TM
Teemu	Salonen	242	2000-07-16	160700A511E	Lohja	TM
Mikko	Timonen	218	1960-06-01	010660-725B	Tampere	TM
Vesa	Risikko	259	1992-01-30	300192-4060	Tampere	TM
Kaisa	Tammi	33	1989-04-08	080489-537N	Vantaa	TM
Teemu	Virtanen	6	1970-12-20	201270-9454	Porvoo	TM
Liisa	Niinimaa	1	1976-08-20	200876-927V	Rovaniemi	TM
Vesa	Timonen	141	1983-10-02	021083-7277	Porvoo	TM
Liisa	Risikko	147	1975-12-01	011275-720W	Kerava	TM
Sonja	Tammi	110	1963-06-06	060663-855J	Turku	TM
Sonja	Noronen	72	1974-02-19	190274-508B	Jyväskylä	TM
Mikko	Kivi	15	1986-10-09	091086-888P	Vantaa	TM
Teemu	Kivi	85	1996-06-17	170696-413W	Porvoo	TM
Jorma	Salonen	354	2000-01-28	280100A1933	Sipoo	TM
Sonja	Koivisto	204	1992-06-11	110692-2004	Rovaniemi	TM
Mikko	Risikko	62	1966-07-28	280766-522K	Helsinki	TM
Liisa	Risikko	147	1975-12-01	011275-720W	Kerava	TM
Mikko	Salonen	249	1993-10-10	101093-007R	Sipoo	TM
Jorma	Noronen	182	1993-09-19	190993-676A	Espoo	TM
Viivi	Virtanen	433	1986-04-07	070486-762H	Rovaniemi	TM

Kuva 6. Näyte generoidusta datasta. [8.]

Generoinnin onnistuminen voidaan tarkistaa menemällä kohdepalvelimelle ja antamalla sql-komennon SELECT * FROM PERSONS, jolloin kaikki 500 riviä generoitua dataa tulee käyttäjän ruudulle.

5.5.3 MAGAZINES-taulu

PERSONS-taulun generoinnin jälkeen tarvitaan ORDERS- ja MAGAZINES-tauluille dataa. Järjestys datan generoinnille on katsottava riippuvuuksien mukaan. ORDERS-taululla on riippuvuuksia tauluille PERSONS ja MAGAZINES, joten on hyvä generoida dataa ensin tauluun MAGAZINES. Molemmille tauluille on kuitenkin ensin tehtävä data pool, joka pitää sisällään näiden taulujen datan generoinnin säännöt. Nämä kaikki kolme data poolia oltaisiin voitu tehdä samalla kertaa, mutta selvyyden vuoksi tehtiin ensin vain yksi data pool. Data poolien tekeminen tauluille MAGAZINES ja ORDERS tapahtuu samalla tavalla kuin aikaisemmin tehdyn PERSONS-taulun Data pool. Kaikki kolme data poolia täytyy kuitenkin olla saman Data Setin alla, joka esimerkkitapauksessa nimettiin 2Data. Kun data poolit on tehty ja rekisteröinnin kautta myös skeemat on saatu näkyviin, voidaan aloitaa sääntöjen rakentaminen. [8.]

Taulun MAGAZINES -sääntöjen tekeminen menee samalla kaavalla kun PERSONS -taulussa, mutta arvot ja funktiot ovat erilaisia. Kun säännöt on saatu valmiiksi, ei ole tarkoitus tehdä 500 riviä dataa, vaan teettää dataa olemassa olevan tuotemäärän mukaisesti. Esimerkkitapauksessa on tuotteita yhdeksän kappaletta, joten myös generoitua dataa tarvitaan yhdeksän riviä ja julkaisussa tarvitaan yhdeksän toistoa. [8.]

ProductID

Yksilöivänä tietona tuotteilla on tuotteen ID, joka on myös taulun MAGAZINES perusavain. ID voitiin tuottaa samalla tavalla kun tuotettiin taulun PERSONS ID -säännöt eli muuttuja-arvolla Next.

ProductName

Tuotteen nimiä varten tarvitaan uusi seed-lista, joka sisältää halutut tuotteet. Tässä esimerkkitapauksessa liitetään listaan yhdeksän eri tuotetta. Sääntö rakennetaan niin, ettei samaa tuotetta tule listasta kahta kertaa. Tämä saadaan aikaan funktiolla @seqlov, joka ottaa listasta seuraavan arvon jokaisella toistolla. On huomioitava, että tätä funktiota käytettäessä toistoja voi olla enintään yhtä monta, kuin on tuotteita.

Publisher

Julkaisijoita varten tarvitaan uusi seed-lista tai järkevämpi tapa olisi lisätä julkaisijat osaksi tuotteiden nimiä eli samaan seed-listaan. Esimerkkitapauksessa kuitenkin päädyttiin tekemään uusi lista, jossa on viisi julkaisijaa. Kun jokaista tuotetta tulee vain yksi kappale, voidaan julkaisijat asettaa tuotteille tuotekohtaisesti. Esimerkkitapauksessa tuotteet eivät vastaa oikeita julkaisijoitaan, koska se ei ole työn kannalta merkittävää. [8.]

Date

Päiväys havainnollistaa tässä tarkoituksessa vain tietoa siitä, milloin tuote on tullut valikoimaan. Päiväys voi tällöin olla ihan mikä tahansa päivämäärä, ja tämän säännön rakentamiseen käytämme samaa funktiota kun PERSONS-taulun Birth-kentän generoinnissa käytimme. Esimerkkitapauksessa toteutetaan funktion @randdate ympärille rakennettu kokonaisuus. Pienennämme kuitenkin skaalaa hieman pienemmäksi verrattuna edelliseen saman-funktion toiminnallisuuteen, eli arvoksi 01.01.2010 - 31.12.2017. [9.]

5.5.4 ORDERS-taulu

Kun taulut PERSONS ja MAGAZINES ovat valmiina, voidaan tehdä säännöt taulun ORDERS -kentille. ORDERS-taulun kentillä on yhden perusavaimen lisäksi kaksi viiteavainta, jotka on otettava huomioon sääntöjä tehtäessä. Kun säännöt saadaan kirjoitettu valmiiksi, julkaistaan ne data targetille samalla tavalla, kun edellisissäkin tauluissa oli tehty. Ohjelmassa olisi mahdollista tehdä tilauksia uskottavasti jokaista henkilöä kohden kattaen kaikki yhdeksän tuotetta. Tällöin toistoja pitäisi olla 4500, mutta esimerkkitapauksessa riittää hyvin vähempi määrä kuten 100. Tällöin asiakkailla saattaa olla tilaustiedot järjestelmässä, mutta ei voimassa olevia tilauksia. [8.]
OrderID

Rakennetaan sääntö ID -arvolle käyttäen Next-parametria alkaen luvusta yksi samalla tavalla, kun aikaisemmissa vaiheissakin toteutettiin. Tämä arvo on määritetty ORDERS-taulun perusavaimeksi ja määritys vaatii sen olevan uniikki.

OrderNumber

Tilausnumero tarvitaan tuottamaan ORDERS-tauluun yhteneväisyyttä tilauksiin. Jaottelu tilauksille tehdään omalla ID-arvolla, mutta kuitenkin voidaan linkittää useiden tuotteiden tilaukset yhteen numeroon. Ehto tälle numerolle on oltava sama tilauksissa, joissa useampi tuote tulee samalla transaktiolla samalle henkilölle. Käytännössä ei siitä sisällytetä esimerkkinä tähän esimerkkitapaukseen, mutta tietokantarakenne antaisi sille mahdollisuuden. [8.]

PersonID

Henkilön ID-arvo on viiteavain, joka osoittaa taulun PERSONS kenttään ID. Tämä tuo mukanaan ehtoja kuten sen, ettei PersonID voi olla suurempi määrä kuin PERSON -taulussa on ID-kentän arvoja. Toisena huomioitavana asiana on saman tuotteen tilaukset samalla henkilöllä, joka katsotaan tässä tapauksessa mahdolliseksi. Asiakas määritellään voivan ostaa tilauksen Helsingin Sanomista sekä itselleen, että esimerkiksi äidilleen. Esimerkkitapauksessa päädytään käyttämään satunnaisesti valittua arvoa yhden ja viidensadan väliltä.

ProductID

Tuotteen ID-tunnus on myös viiteavain, joka osoittaa taulun MAGAZINES -kenttään ProductID. Tuotteita työssä on yhdeksän kappaletta, joka asettaa vaatimuksen siitä, että esimerkiksi ProductID kymmenen ja sitä suuremmat luvut eivät kelpaa. Esimerkkitapauksessa käytetään @randvalue-funktiota yhden ja yhdeksän välillä olevilla arvoilla. [8.]

Todennus

Kun kaikkiin tauluihin on saatu generoitua halutut tiedot, voidaan tarkistaa vielä niiden olemassaolo Data Target -symbolin takaa. SQL-komennoksi annettiin tässä esimerkkitapauksessa. SELECT PERSONS.FirstName, PERSONS.LastName, PERSONS.ID, PERSONS.Birth, PERSONS.FinnishID, PERSONS.City, MAGAZINES.ProductName FROM ORDERS full join MAGAZINES on MAGAZINES.Productid=ORDERS.Productid full join PERSONS on PERSONS.ID=ORDERS.PersonID where MAGAZINES.ProductName='HS' or MAGAZINES.ProductName='TM' or MAGAZINES.ProductName='Kauppalehti';, joka antaa käyttäjän näytölle tietoja kaikista kolmesta taulusta ja näin pystytään toteamaan, että viite-eheys on ehjä.

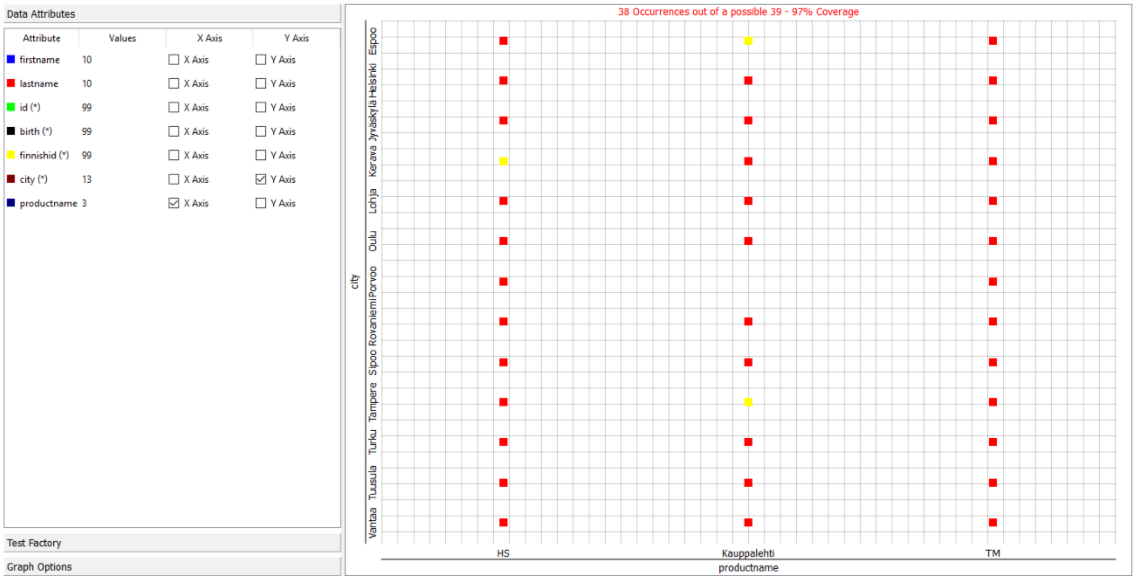
5.6 Datan Visualisointi

Kun työssä on tarkastettu viite-eheys generoiduista tiedoista, voidaan visualisoida sitä dataa käyttäen GT Data Visualizer-ohjelmaa. Haku näyttää kaiken datan, joissa tuotteena on Helsingin Sanomat, Tekniikan Maailma tai Kauppalehti. Näkyväksi kyseisistä tiedoista tulee etunimi, sukunimi, id, syntymäaika, Henkilötunnus, kaupunki ja tuotteen nimi. Avatessasi GT Data Visualizer -ohjelman, tulee auki kolme eri välilehteä, joista yksi on vakio opetussivu. Tässä esimerkkitapauksessa ei käydä opetussivua tämän tarkemmin läpi. [9.]

Datan kattavuus

Kun tarkastellaan kahta jäljelle jäänyttä välilehteä huomataan, että molemmissa on listattu vasemmalle puolelle SQL-komennossa valitut tiedot, joita käyttämällä voidaan rajata visualisoitua dataa. Välilehdet jakautuvat kahteen erilaiseen näkymään ja käyttötarkoitukseen. Nämä kyseiset välilehdet ovat hyödyllisimpiä datan visualisoinnin työkaluja, mutta tässä esimerkkitapauksessa käydään niistä vain toinen johtuen virtuaalikoneen heikosta grafiikkakortista. Esimerkkitapauksessa käydään läpi Spot graph -ominaisuus, mutta P-coords -ominaisuus on jätetään tästä työstä pois. Lyhyesti selitettynä P-Coords kuvaa datan kattavuutta rinnakkaisten viivojen avulla. Mitä paksumpi viiva, sitä kattavampi data näiden valittujen kenttien välillä on. [9.]

Oikealla puolella on kenttiä, joita voi valita joko x- tai y- akselille. Näiden valintojen avulla voidaan toteuttaa pistekartan siitä datasta, jota on esimerkkitapauksessa tehty. Punaiset neliöt kuvaavat heikkoa dataotantaa eli kyseisen variaation toistuu alle viidessä rivissä dataa. Keltaiset neliöt kuvaavat dataa, jonka otanta on enemmän kuin viisi, mutta vähemmän kuin 19 riviä. Vihreät neliöt kuvaavat tarpeeksi kattavaa otantaa datasta eli enemmän kuin 19 riviä. Otannan määrityksiä voi säätää tarpeen mukaan. Esimerkkitapauksessa vakioasetukset ovat riittäviä ja kentistä valitaan tuotteiden nimet ja kaupungit. Pistekartasta voi huomata, että tiedot näillä ehdoilla kattavat 97 prosenttia kaikista variaatioista ja kaikki ehtoon kuuluvat rivit konkreettisesti näkyville saimme valitsemalla halutun värisen neliön.

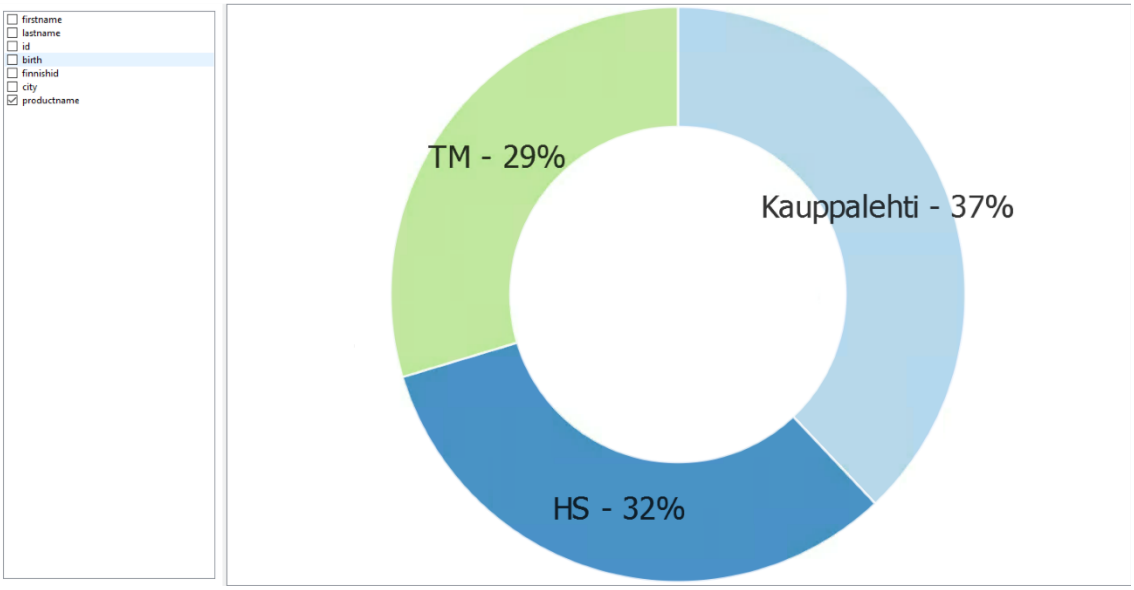


Kuva 7. Datan kattavuus, kun valittuna on kentät City ja ProductName.

Kuva 7 selventää hyvin sen, ettei sadalla tilauksella saada kovin monen rivin kattavuutta aikaiseksi. Tärkeimpänä huomiona oli kuitenkin se, ettei kattavuutta ollut Porvoossa tilatuille Kauppalehdelle ja saavuttaaksemme täysi kattavuus testeille olisi saavutettava kaikki variaatiot. Valitsemalla Test Factory -sivun, voidaan selvittää suoraan, mikä osa datan variaatiosta puuttuu. Kun puuttuvat variaatiot on selvitetty, GT Data Visualizer näyttää täytetyn kolon vihreällä neliöllä, ja se neliö pitää sisällään kaikki variaatiot puuttuvaan kohtaan. Seuraavaksi ladataan ehdotetut variaatiot ulos valitsemalla Export Values, joka antaa mahdollisuuden tallentaa ehdot CSV-tiedostoon. Tätä kyseistä CSV-tiedostoa voimme käyttää esimerkiksi GT DataMaker -ohjelmassa datan generoinnin tukena. [9.]

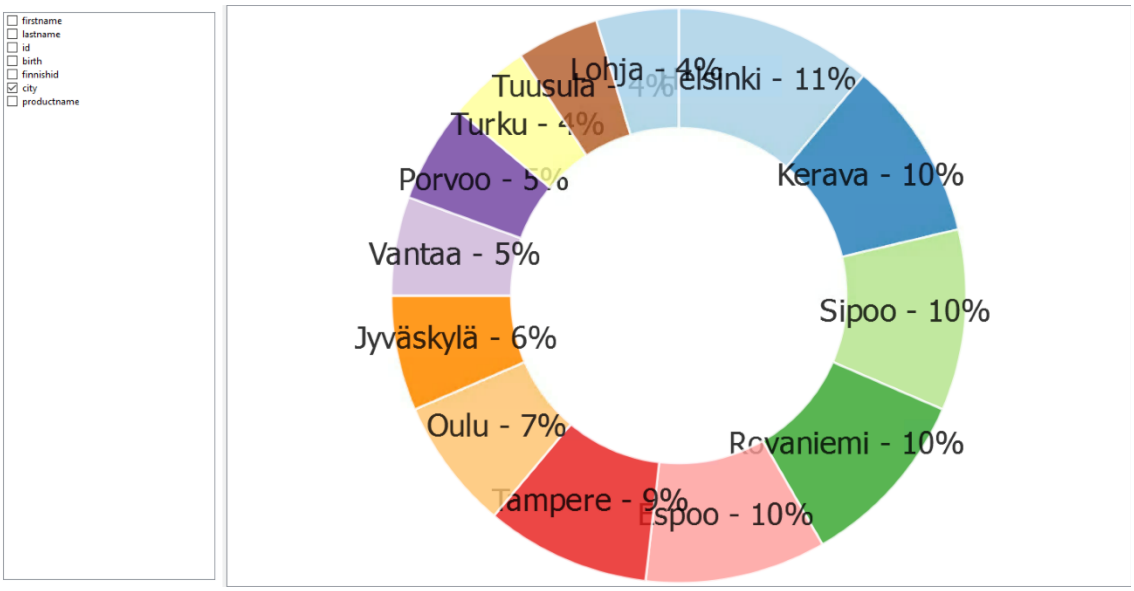
Datan visualisointi käyttäen eri graafeja

Kahden vakiona tarjottavan visualisointi tavan lisäksi voidaan lisätä myös muunkaltaisia graafeja. Tämä tapahtuu ohjelman osiosta Add Graph, joka avaa ikkunan tarjolla olevista graafeista. Vakioiden lisäksi valittavissa on pylväsdiagrammi, sektoridiagrammi, säde- ja suorakulmainen esitys funktionaaliselle kattavuudelle ja perinteinen datataulu. Tässä esimerkkitapauksessa näytetään muutama esimerkki sektoridiagrammista ja siitä, millainen vaikutus datan visualisointiin kyseisellä graafilla on.



Kuva 8. Sektoridiagrammi valituista tuotteista. [9.]

Sektoridiagrammi näyttää kenttien tietojen kattavuutta prosentteina ja verraten oman kenttänsä muihin tietoihin, jolloin yleisesti vähemmällä datalla varustetut kentät näyttävät järkevämpiä sektoridiagrammeja. Ensimmäisessä testissä valitaan kentäksi tuotteen nimi, joka jakoi tiedot kolmen aikaisemmin valitun tuotteen Helsingin Sanomien, Kauppalehden ja Tekniikan Maailman mukaisesti (kuva 8). [9.]



Kuva 9. Sektoridiagrammi haun kaupungeista. [9.]

Toisessa testissä valitaan hieman enemmän tietoa sisällään pitävän kentän kaupungit, joka on enemmän diagrammin käytön riskirajoilla. Tietoja on niin paljon, että nimet alkavat mennä päällekkäin. Sektoridiagrammi (kuva 9) kertoo kuitenkin, että datan kattavuus on muita heikompaa Lohjan, Turun ja Tuusulan kohdalla. Kun datan kattavuutta testejä varten halutaan nostaa, olisi järkevää nostaa näiden kaupunkien näkyvyyttä generoitavassa datassa. [9.]

5.7 Datan maskaaminen

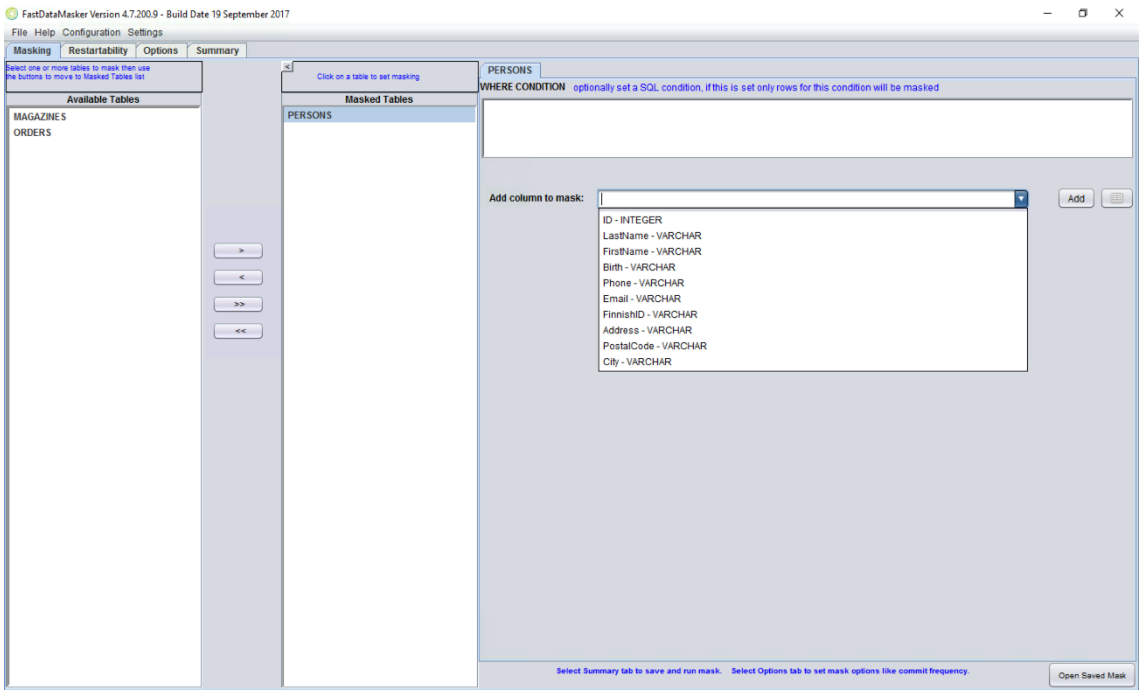
Kun on tarkoituksena datan maskaaminen, on käytettävä ohjelmaa FastDataMasker, joka toimii täysin erillisenä ohjelmana GT DataMaker -ohjelman ulkopuolella. Jos halutaan tehdä kaikki muutokset keskitetysti näiden ohjelmien välillä, olisi koko pakettia hallittava kolmannella ohjelmalla nimeltä Javelin. FastDataMasker-ohjelma olisi mahdollista saada myös täysin erillisenä ohjelmana, ja tämä voisi olla hyvä ratkaisu yrityksille, joilla on kaikki muut osa-alueet kunnossa. [10.]

Kun FastDataMasker -ohjelman avaa ensimmäistä kertaa, on asetettava tietokantojen ja ohjelman välille yhteys. Esimerkkitapauksessa liitettiin ohjelmaan molemmat tietokannat TDMTESTSOURCE ja TDMTESTTARGET. Käyttäjähallinta toimii ohjelmassa tietokantojen oman käyttäjähallinnan kautta, jolloin jos käyttäjällä on pääsy tietokantaan, niin hän pystyy myös maskaamaan kyseistä tietokantaa tällä ohjelmalla. Kun yhteys luodaan onnistuneesti, yhteydestä tulee ohjelman käyttöön myös tiedosto, joka pitää kaikki yhteyden konfiguraatiot sisällään. Kun seuraavalla kerralla haluaa jatkaa maskaamista näihin tietokantoihin, kaikki yhteyden konfiguraatiot ovat jo olemassa. [10.]

Maskaamisen ideana on anonymisoida tuotannon henkilötietoja niin, ettei kyseisestä datasta ole enää mahdollista tunnistaa alkuperäisiä henkilöitä. FastDataMasker haluaa yhdistetyn tietokannan sisältävän tuotannon datat valmiina. Tämä saattaa olla ongelma joissain yrityksissä, koska FastDataMasker pystyy tuomaan dataa näytille yhdistetystä tietokannasta. Mutta ylipäätänsä pystyäkseen maskaamaan dataa, on käyttäjällä oltava tarpeeksi oikeuksia päästä näkemään tuotannon datat muillakin tavoilla. Tähän on muutamia eri ratkaisuja olemassa ja kaksi tietokantayhteyttä on niistä ratkaisuista yksi. Kahden yhteyden välillä voidaan tuotannon tietokannasta hakea tietoa, maskata kyseinen tieto ja siirtää toiseen tietokantaan. Esimerkkitapauksessa tämä ratkaisu toimii yksittäisten rivien kohdalla, mutta kun on tarkoitus maskata useampia rivejä, niin rivien arvot näiden toiminnallisuuksien osalta ovat samat. Esimerkkitapauksessa ei tutkittu asiaa sen syvemmin, mutta selvästi FastDataMasker pystyisi maskaamaan dataa kannasta toiseen myös käyttäjän dataa näkemättä. [10.]

PERSONS-taulu

Valittuaan halutun tietokantayhteyden on valittava maskattavat taulut. FastDataMasker-ohjelma kerää kaikki yhdistetyn tietokannan taulut ikkunan vasempaan reunaa. Kun haluttu taulu on saatu valittua, niin on se siirrettävä maskattavien taulujen puolelle. Tarpeen mukaan maskattavien taulujen puolella voi olla useitakin tauluja valittuna.



Kuva 10. Maskattavien taulujen valitseminen. [10.]

Esimerkkitapauksessa halutaan havainnollistaa todellista tilannetta ja todellisessa tilanteessa vain henkilötiedoilla on tarve maskaukselle. Usein kaikki muu tieto pystytään joko käyttämään suoraan tai helposti generoimaan tyhjästä. Kun valitaan PERSONS-taulu maskausta varten, tulee meille mahdollisuus valita sarakkeita ja vetovalikosta löytyvät kaikki kyseisen taulun sarakkeet. [10.]

ID

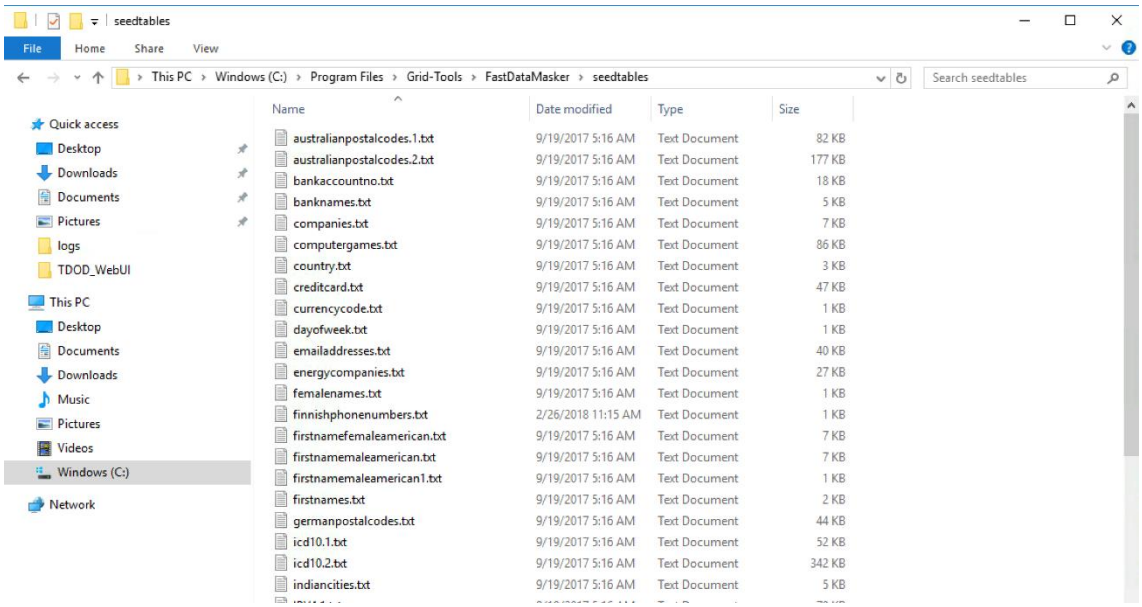
Pääsääntöisesti ID ei tarvitse maskausta, mutta jos halutaan varmistaa onnistunut lopputulos, niin ID voidaan ratkaista maskaus-SEQNUMBER-funktiolla, että ID-luvut kulkevat järjestyksessä. Tällöin ID saa arvot yhdestä eteenpäin niin pitkälle, kun yhdistetyssä tietokannassa on rivejä.

FirstName ja LastName

Nimien suhteen on monia eri vaihtoehtoja toteuttaa maskausta, mutta kaksi suosittua tapaa ovat RANDLOV-funktiolla arvojen haku tiedostosta tai tietokannasta. Kun aikaisemmin mainittiin, kuinka tietokantojen välillä voi siirrellä ja maskata dataa, tietokantojen satunnaiset arvot on juuri se tapa. Esimerkkitapauksessa kiinnitetään enemmän huomiota tiedoston satunnaisiin arvoihin. Tätä varten tehtiin seed-lista, joka rakennetaan hieman eri tavalla kuin aikaisemmissa vaihessa. Kun seed-listat on tehty halutulla tavalla, voi ne liittää datakategorioiksi ja kyseisen kentän maskauksen ehdot ovat valmiit. Ajettaessa tiedot tulevat sattumanvaraisesti riveiltä tehdyn seed-listan sisältä. [10.]

Seed List

Ohjelma itsessään ei tarjoa mahdollisuutta muokata seed-listoja, joten tapa ratkaista tämä ongelma ei ole ihan tavallinen ohjeistus. Seed-listan tekemistä varten on löydettävä paikka CA TDM -asennuksen rakenteesta, josta FastDataMasker ohjelma hakee jo valmiina olevat seed-listansa. Esimerkkitapauksessa kyseinen polku löytyy etsimällä valmiin seed-listan nimellä vastaavia listoja palvelimen juuritasolta.



Kuva 11. Listojen sijainti FastDataMasker -ohjelmassa.

Lopullinen polku on C:/Program Files/Grid-Tools/FastDataMasker/seedtables ja kyseinen polku toimii myös esimerkkitapauksen ulkopuolella, jos CA TDM-ohjelman asennus on tehty käyttäen vakiona annetuja polkuja. Itse listojen tekeminen ei ole monimutkaista. Kun tehdään edellä mainittuun polkuun uuden tekstitiedoston, annataan sille kuvaava nimi ja täytetään se halutuilla tiedoilla niin, että yksi tieto on aina yhdellä rivillä.

Phone

Puhelinnumeron kohdalla viedään ehtoa hieman ohi maskauksen idean mutta tuodaan samalla uusi näkökulma testidatan valmisteluun. Jos testauksessa tarvitaan toimivaa puhelinnumeroa, ei tietenkään voi käyttää asiakkaiden omia numeroita. Tarvitaan testaukseen tarkoitettuja numeroita, ja jos numeroita olisi enemmän kuin yksi, niin voidaan käyttää seed-listoja niiden asettamiseksi. Esimerkkitapauksessa havainnollistetaan tilannetta, jossa on yksi testauksen puhelinnumero, joka annettiin kaikkien tietueisiin. Käytetään FIXED-funktiota, joka antaa asettaa haluamansa arvon kaikille. Esimerkkitapauksessa arvo oli +358401112222. [10.]

Email

Sähköpostin kanssa on samanlainen tilanne, kuin oli puhelinnumeron kanssa. Ei ole hyvä käyttää oikeita asiakkaiden osoitteita testauksessa ja siksi on vaihdettava sähköpostiksi testaukseen tarkoitetut sähköpostiosoitteet. Seed-listastan ja FIXED-funktion väliltä valittiin tässä esimerkkitapauksessa jälkimmäinen vaihtoehto ja arvoksi asetimme dummy.email@tdm.fi.

FinnishID

Suomalaisia henkilötunnuksia varten FastDataMasker-ohjelmassa on valmis funktio UNIQUEFINNISHID, joka generoi varmasti käyttämättömiä henkilötunnuksia. Soveltaminen henkilötunnuksen kanssa on vaarallista, ja sen suhteen on oltava varovainen. Toiminnassa oleva henkilötunnus lasketaan heti yksilöivänä henkilötietona ja sellaisen pitäminen testiympäristössä ei ole sopivaa eikä laillista. [10.]

Address, City ja Post

Osoitteiden muutokset eivät yleensä ole kovinkaan tarpeellisia maskaamisen näkökulmasta. Osoite itsessään ei yleisesti ole yksilöivää tietoa, mutta se saattaa muodostaa henkilötiedon yhdessä muiden tietojen kanssa. Yleisesti pelkkä kadun muuttaminen riittää osoitetietojen maskauksessa, jolloin kaupungin ja postinumeron muuttaminen ei ole enään tarpeellista. Tässä esimerkkitapauksessa käydään kuitenkin kaikille kentille ratkaisut. Kadunnimet ja kaupungit vaihdetaan käyttäen RANDLOV-funktiota ja etukäteen tehtyjä seed-listoja. Postinumeroiden kohdalla kokeiltiin valmista funktiota USZIP, joka antaa arvoksi amerikkalaisia postinumeroita. [10.]

Julkaisu

Kun kaikki maskauksen ehdot on päätetty, voidaan siirtyä summary-välilehdelle, joka näyttää yhteenvedon tekemistä ehdoista. Tässä vaiheessa olisi vielä mahdollista tehdä viimeisen hetken muutoksia, mutta jos muutoksia ei ole tarve tehdä, niin voimme hyväksyä valinnat valinnalla Save & Run Mask. [10.]

Taulukko 1. Yhteenvetotaulukko annetusta maskaus ehdoista.

Masking | Restartability | Options | Summary

	Table	Column	Function	Parm1	Parm2	Parm3
1	PERSONS	ID	SEQNUMBER			
2	PERSONS	FirstName	RANDLOV	finnish_firstnames.txt		
3	PERSONS	LastName	RANDLOV	finnish_lastname.txt		
4	PERSONS	Phone	FIXED	+3581112222		
5	PERSONS	Email	FIXED	dummy.email@tdm.fi		
6	PERSONS	Address	RANDLOV	Finnishstreetnames.txt		
7	PERSONS	City	RANDLOV	Finnishcity.txt		
8	PERSONS	FinnishID	UNIQUEFINNISHID			
9	PERSONS	PostalCode	USZIP			

Tallennusominaisuus tallentaa ehdot CSV-tiedostoksi, jolloin itse maskaus alkaa. Riippuen kohdetietokannan koosta maskaus voi olla joko nopeaa tai hieman vähemmän nopeaa, mutta tällöin puhutaan tuhansista riveistä.

6 Tulos, ratkaisu ja pohdinta

6.1 Tulos

Esimerkkitapauksessa suoritettiin datan generointi, datan maskaus ja datan visualisointi yksinkertaisimmillaan. Tuloksena saatiin tietokanta, jossa oli generoitua dataa sekä toinen tietokanta, joka sisälsi maskattua dataa. Työssä tutustuttiin myös datan visualisointiin, josta opittiin selvittämään datan kattavuudessa olevia puutoksia. Selvitettiin myös, kuinka puuttuva tieto saadaan tietokantaan testausta varten. Työssä tutustuttiin myös GT Data Visualizer -ohjelman tarjoamiin kaavioihin, joilla pystyimme tutkimaan datan kattavuutta kentittäin.

Taulukko 2. TDMTESTSOURCE-instanssin generoitu datanäyte

All 4 rows returned — MS Sans Serif

Id	Lastname	Firstname	Birth	Phone	Email	Finnishid	Address	Postalcode	City
1	Niinimaa	Liisa	1976-08-20	+358502394106	Liisa.Niinimaa@tdm.fi	200876-927V	Hakkukuja	05381	Rovaniemi
2	Salonen	Viivi	1960-07-14	+358400940395	Viivi.Salonen@tdm.fi	140760-101D	Jaalantie	01842	Helsinki
3	Noronen	Vesa	1974-09-13	+358402414511	Vesa.Noronen@tdm.fi	130974-2261	Haasia	06807	Oulu
4	Salonen	Kaisa	1966-12-27	+358409583220	Kaisa.Salonen@tdm.fi	271266-860V	Salpaharjuntie	02325	Sipoo

Näytteet haettiin komennolla SELECT * FROM PERSONS WHERE ID < 5 (taulukko 2 Ja taulukko 3).

Taulukko 3. TDMTESTTARGET instanssin maskattu data näyte

All 4 rows returned — MS Sans Serif

Id	Lastname	Firstname	Birth	Phone	Email	Finnishid	Address	Postalcode	City
1	Kivi	Armas	1984-09-28	+3581112222	dummy.email@tdm.fi	010120-001M	Katulankuja	99352	Turku
2	Tihinen	Heli	1990-02-13	+3581112222	dummy.email@tdm.fi	010120-002N	Sannankuja	65129	Espoo
3	Vaarama	Eemeli	1970-02-17	+3581112222	dummy.email@tdm.fi	010120-003P	Talkokatu	47289	Helsinki
4	Kallio	Toivo	1961-06-18	+3581112222	dummy.email@tdm.fi	010120-004R	Talkokatu	11452	Pori

Esimerkkitapauksessa käytiin datan generointia enemmän tuotannon näkökulmasta, mikä näkyy taulukossa 2. Työssä maskattiin dataa tuotannon datasta testauskäyttöön. Kun katsotaan näitä kahta kuvaa, voimme todeta, ettei maskatusta datasta voida tunnistaa alkuperäisiä henkilöä. Todellisessa tilanteessa näin rankka maskaaminen ei ole tarpeellista, mutta yksittäisiä kenttiä pystytään tarkastelemaan oikeaa tilannetta silmällä pitäen.

6.2 Ratkaisu

Lopputyön teorian ja esimerkkitapauksen tuloksen perusteella voidaan nähdä GDPR-vaatimuksien mukaisen testidatan hallinnan mahdollisena toteuttaa. Kun yritys ratkaisee ongelmaa testidatan hallinnassansa, on huomioitava seuraavat parannukset. Yrityksen tietokantojen rakenne tulisi täyttää vähintään minimivaatimukset. Tämä pitää sisällään tuotannon tietokantojen käyttäjähallinnan, tuotannon tietokannasta otettavat varmuuskopiot, testidatana ei käytetä tuotannon dataa ja yrityksellä on palkattu henkilö vastuussa yrityksen datasta ja tietokannoista.

Seuraavana parannuksena tulisi huomioida GDPR:n vaatimukset. Yrityksellä täytyy olla tietosuojavaltuutettu, jonka tehtävänä on huolehtia yrityksen GDPR vaatimuksien noudattamisista ja tietosuojatarkastusten järjestämisestä. Osalla yrityksistä on mahdollisuus sisällyttää kyseisen henkilön työt jo olemassa olevalle työntekijälle, kunhan asiat tulee hoidettua.

Viimeisenä parannuksena tulee testidatan hallinnan toteuttaminen, joka itsessään on enemmän kulttuurimuutos. Tämän ongelman ratkaisemiseksi on mahdollista rakentaa skriptaamalla oma työkalupakki, jolla testidataa hallitaan tai vaihtoehtoisesti yritys voi hankkia jonkun markkinoilla olevista kaupallisista tuotteista. Havainnollistavassa esimerkkitapauksessa kävimme läpi yhden kaupallisen tuotteen tarjoamat mahdollisuudet.

6.3 Pohdinta

Esimerkkitapauksessa kävimme vain yksinkertaisen osan kyseisen työkalun kyvyistä toteuttaa testidatan hallintaa, joten työ sellaisenaan ei anna kuin pientä suuntaa todelliselle testidatan hallinnalle. Yrityksen tarve vaatii todennäköisesti myös ominaisuuksia, jotka jätimme esimerkkitapauksesta pois. Työ tehtiin vastaamaan yritysmaailmassa esitettävien esiselvitystöiden pituutta ja kattavuutta. Esiselvitystyön tarkoituksena on antaa alustavaa kuvaa tavoista ja tuotteista muutaman tunnin tapaamisessa.

Ennen GDPR -asetuksen voimaantuloa emme voi tietää, millä tarkkuudella lakia tulkitaan. Monet yritykset odottavat ennakkotapausta GDPR-asetuksen rikkomuksista ja sen mukana tulevaa ensimmäistä rangaistusta. Kahden vuoden siirtymäajasta huolimatta yritykset ovat heränneet näiden ongelmien ratkaisemiseen viimeisen puolen vuoden aikana, eikä kaikilla yrityksillä näin ollen ole mahdollista toteuttaa GDPR-vaatimuksien kattavaa toimintaan ennen asetuksen voimaantuloa 25. toukokuuta 2018.

Lähteet

1 History of testing. Verkkoaineisto. <https://en.wikiversity.org/wiki/Software_testing/History_of_testing> Luettu 03.08.2017.

2 Test Data Management explained in 6 minutes (ReliableInformationSystems.tv). 2016. Verkkoaineisto. <https://www.youtube.com/watch?v=wxFxvh21mXQ> Katsottu 27.10.2017.

3 GDPR EU:n tietosuoja asetus, Havain. 2017. Verkkoaineisto. <https://www.havain.fi/nyt-puhuttaa-gdpr-eu-tietosuoja-asetus/> Luettu 18.11.2017.

4 GDPR Portal. Verkkoaineisto. <http://www.eugdpr.org/> Luettu 18.11.2017.

5 Maksimainen, Olli. Mitä sinun tulee tietää GDPR-tietosuoja-asetuksesta? 2017. Verkkoaineisto. <https://blog.crasman.fi/mita-sinun-tulee-tietaa-gdpr-tietosuoja-asetuksesta > Luettu 18.11.2017.

6 Test Data Manager. Verkkoaineisto. <https://www.ca.com/us/products/ca-test-data-manager.html> Luettu 21.03.2018.

7 CA TDM Project Setup & Walkthrough. CA Technologies. 2017. Verkkoaineisto. <https://www.youtube.com/watch?v=1a39XDr2cG0> Katsottu 02.04.2018.

8 CA TDM Data Generation - 101 guide. CA Technologies. 2017. Verkkoaineisto. <https://www.youtube.com/watch?v=2r3wiBfJ0po> Katsottu 03.04.2018.

9 CA Test Data Visualizer - the basics. CA Technologies. 2017. Verkkoaineisto. <https://www.youtube.com/watch?v=b8iW2Hd96qM&list=PLO7SodxCJyn5McP0qjyEY1nIQ2hy1A_Nh&index=5> Katsottu 04.04.2018.

10 CA TDM Data Masking - 101 guide. CA Technologies. 2017. Verkkoaineisto. <https://www.youtube.com/watch?v=H_3tAsZt_nY&t=759s> Katsottu 05.04.2018.

11 About us. CA Technologies. Verkkoaineisto. <https://www.ca.com/us/company/about-us.html?intcmp=headernav> Katsottu 05.04.2018.

12 Sisäinen kouluttautuminen, Eficode Oy, 03.04.2017.

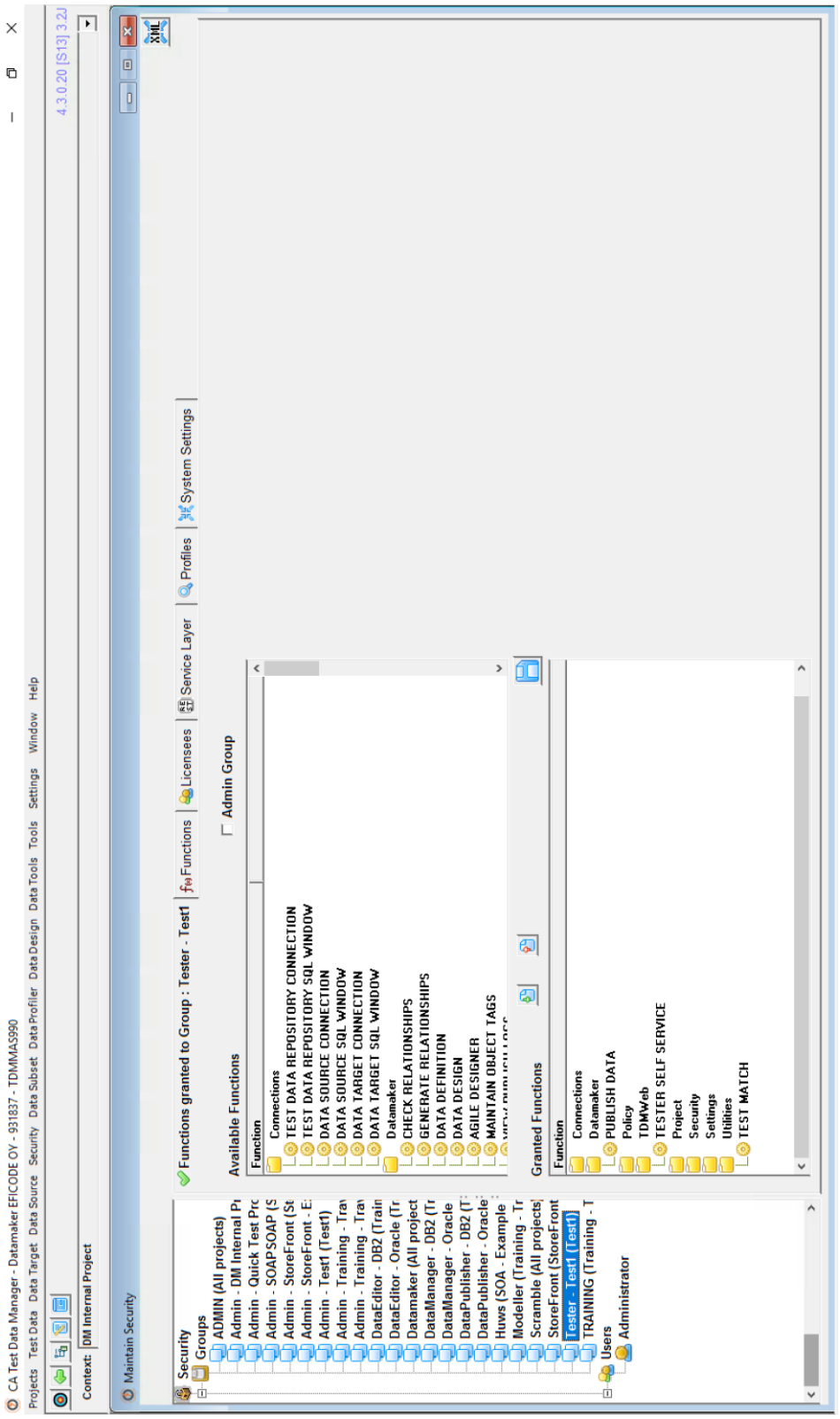
Tietokannan SQL-skripti

```
CREATE TABLE PERSONS (
   ID int NOT NULL PRIMARY KEY,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Birth varchar(255),
   Phone varchar(255),
   Email varchar(255),
   FinnishID varchar(12),
   Address varchar(255),
   PostalCode varchar(5),
   City varchar(255)
);


CREATE TABLE MAGAZINES (
    ProductID int NOT NULL PRIMARY KEY,
    ProductName varchar(255) NOT NULL,
    Publisher varchar(255),
    Date varchar(60)
);

CREATE TABLE ORDERS (
    OrderID int NOT NULL PRIMARY KEY,
    ProductID int NOT NULL,
    PersonID int FOREIGN KEY REFERENCES PERSONS(ID),
    ProductID int FOREIGN KEY REFERENCES MAGAZINES(ProductID)
);
```

Käyttäjähallinnan käyttöliittymäkuva



Projektin luonnin käyttöliittymäkuva